Universidad Autónoma de Madrid
Escuela politécnica superior

Máster Oficial EPS: Bioinformática y Biología Computacional

Trabajo Fin de Máster

DESARROLLO DE UNA WEB DE ANÁLISIS DE DATOS NGS PARA LA DETECCIÓN DE MOSAICISMO GENÉTICO EN ENFERMEDADES RARAS Y EXPERIMENTOS DE EDICIÓN GENÓMICA (CRISPR-CAS9).

Autor: Sergio Fernández Peñalver
Tutor 1: Miguel Angel Moreno Pelayo
Tutor 2: Val Fernández Lanza
Ponente: Luis del Peso Ovalle

FEB 2019

DESARROLLO DE UNA WEB DE ANÁLISIS DE DATOS NGS PARA LA DETECCIÓN DE MOSAICISMO GENÉTICO EN ENFERMEDADES RARAS Y EXPERIMENTOS DE EDICIÓN GENÓMICA (CRISPR-CAS9).

Autor: Sergio Fernández Peñalver
Tutor 1: Miguel Angel Moreno Pelayo
Tutor 2: Val Fernández Lanza
Ponente: Luis del Peso Ovalle

Hospital Ramón y Cajal
IRYCIS-CIBERER-RAREGENOMICS
Servicio de Genética
Universidad Autónoma de Madrid
Escuela Politécnica Superior
Departamento de Informática
FEB 2019

Resumen

Resumen

Este proyecto responde a la creciente necesidad de analizar y visualizar datos de secuenciación de nueva generación (NGS) producidos a través de experimentos de edición genética. Particularmente para aquellos utilizando tecnología de CRISPR-CAS9 [1]. Actualmente existen múltiples herramientas online que llevan a cabo cierto nivel de análisis de estos datos [2, 3], sin embargo proporcionan información aglutinada en forma de gráficos de in-dels (inserciones y deleciones) o gráficos de tarta sobresimplificados. No proporcionan una caracterización específica de los efectos de mosaicismo que pueden surgir a raíz de este tipo de experimentos de edición, efectos que pueden resultar determinantes para concluir que un experimento se ha llevado a cabo con éxito o no [4, 5, 6, 7, 8, 9]. Por ello el objetivo de esta aplicación es el de analizar y visualizar estos datos de secuenciación masiva de una manera sencilla y comprensible incluso para aquellos sin gran conocimiento bioinformático.

Para llevar a cabo el proyecto se hizo uso del popular entorno de R [10] en un sistema operativo Linux. R tiene disponible una potente herramienta de desarrollo de páginas web, ShinyR [11]. Mediante su uso uno puede combinar las increíbles capacidades de procesado de datos de R con el desarrollo de una interfaz intuitiva y adaptable.

El desarrollo se subdividió en dos fases principales: el análisis y el agrupamiento de los datos; y el procesado y visualización de estos. El análisis de los datos no es distinto de el procesado llevado a cabo para la identificación de variantes [12]. Comprobaciones de calidad, *trimming* de los adaptadores y primers, filtrado por calidad y tamaño y *joining* de secuencias en caso de estar trabajando con datos en *paired-end* [13]. Tras este procesamiento comienza el agrupamiento de los datos. En un proceso de identificación de variantes ahora precederíamos con un mapeo y usaríamos un llamador de variantes [14], sin embargo en este caso simplemente comparamos cada una de las secuencias entre ellas y las agrupamos según su similitud. Por cada grupo se determina una secuencia representativa o consenso y esta se la alinea frente a una secuencia de referencia. Donde la secuencia de referencia es la secuencia mayoritaria de la muestra sin editar. De esta manera si parte de las secuencias han sido mutadas mediante CRISPR-CAS9 , el alineamiento las identificaría al detectar in-dels (inserciones-deleciones) o cambios de base con respecto a la secuencia representativa. Ya que también somos capaces de asignar a cada grupo de secuencias un valor de *Abundancia* podemos determinar qué porcentaje de las secuencias han sido correctamente editadas frente a las que no.

El proceso por el que CRISPR-CAS9 introduce mutaciones discretas es también susceptible de producir mutaciones aleatorias en posiciones aleatorias [15]. Estas mutaciones aleatorias pueden ser más o menos abundantes, y pueden tener un efecto patogénico o no, motivo por el cual su visualización a nivel de secuencia es necesaria.

La aplicación proporciona muchas opciones a la hora de visualizar los resultados provenientes del análisis. Si el usuario proporciona un secuencia *Target* la aplicación llevará a cabo una búsqueda a través de todos los grupos buscandola e indicará al usuario qué grupo es el más parecido a ella. También facilita el análisis de las secuencias mediante BLASTn [16], la descarga de archi-

vos de alineamiento y los archivos FASTA de todas o algunas de las secuencias representativas, múltiples gráficos e informe customizable que permite al usuario seleccionar la información que él o ella estime relevante.

Palabras Clave

- **Secuenciación de nueva generación:** También conocido como secuenciación másiva, conjunto de métodos y técnicas para la lectura de cadenas de DNA proveniente de muestras biológicas.
- **CRISPR-CAS9 :** Repeticiones Palindrómicas Cortas Agrupadas y Regularmente Interespaciadas con proteina asociada 9. Unión de bases nucleotidas y una proteina capaz de identificar secciones específicas de DNA y llevar a cabo cortes en la doble cadena.
- **Mosaicismo genético:** Condición genética en la que en un mismo organismo coexisten varias células o tejidos con distinto genotipo.
- **Identificación de variantes:** Proceso por el cual, partiendo de unos datos crudos, se identifican las bases nucleotidas de una cadena de DNA que varían en respecto a lo esperado o lo standard, estas variantes comunmente se denominan mutaciones.
- **Variante patogénica:** Mutación que se traduce a una perdida de rendimiento o perdida de función de un gen necesario.
- ***paired-end*:** Método de lectura de secuencias de DNA en la que se leen paralelamente ambas cadenas, de 5' a 3' y su reversa complementaria, originando dos archivos FASTQ para la misma secuencia.

Abstract

Gene-editing is on the rise, and as such, so is the necessity of creating analysis and visualization tools capable of keeping up with the demands of investigators without the bioinformatic background required to deal with pipelines and standalone programs. This web application attempts to integrate the analysis and reporting of New Generation Sequencing (NGS) data that stems from gene-editing protocols, making use for example of Clustered Regularly Interspaced Short Palindromic Repeats-Associated Protein 9 (CRISPR-CAS9). An analysis pipeline was developed, and reporting of the data was automated, making the use of the application extremely simple, intuitive and accesible to users of any background. To this end the application consists of 3 main modules; the analysis and clustering module, formed by a pipeline for NGS data analysis, the visualization module, developed in the ShinyR environment, and a custom automatic report generation tool, developed in Rmarkdown.

Summary

This project responds to the increasing neccesity to analise and visualise new generation sequencing (NGS) data produced by gene editing experiments. Particularly for those using CRISPR-CAS9 technology [1]. Currently multiple online tools provide with some analysis of these data [2, 3], but they provide an aglutination of information in the form of graphs and pie-charts. They fail at providing specific characterisation of the mosaicism effect of these gene-editing protocols, when it is precisely the observation at a sequence level what can be proven deterministic at concluding that an experiment has been developed succesfully [4, 5, 6, 7, 8, 9]. Therefore this aplication's aim is to analyse and visualise this data in an straight forward way, even for those without the bioinformatic knowledge behind them.

To perform this project the popular R [10] environment in-LinuxOS was used. R has available an incredibly powerful web developing tool, ShinyR [11]. With its use one can combine the powerful data processing capabilities of R with the development of an intuitive, adaptive and user friendly interface.

The development was subdivided in two main phases; the analysis and clustering of the data, and the processing and visualization of it. The analysis of the data does not prove very different from the basic analysis any NGS data is subjected to for the identification of variants [12]. Quality check, *trimming* of adaptadores and primers, filtering by quality and minimal length, and *joining* of sequences in case of paired-end data [13]. After this process, starts the *clustering of the data*. While normally for a process of variant identification a mapping and a variant caller would be required to be used [14], here we simply compare every single sequence with each other and distribute them in groups determined by their similarity. For every group a representative/consensus sequence is chosen and this sequence will be aligned against a reference sequence. This reference sequence would be the un-edited sequence that should be found as the most abundant in our sample. This way, if part of the sequences have been mutated by a gene-editing process then the specific changes that were made through the process would be picked up as indels (insertions and deletions) and mismatches during the alignment of the cluster representatives with the reference. Since we are able to asign an *Abundance* variable to every cluster we can identify what percentage of sequences have been mutated and what percentage has remained wild.

It is known that the mutation inducing process of CRISPR-CAS9 produce random out of sight mutations [15]. These random mutations may be less or more abundant and prove insignificant or pathogenic, which is why a visualization of the mutation is provided in the visualization stage.

Many options are provided to the user when the time comes to analyse the results. If the user provided a *Target* sequence the application will look for this sequence within all the clusters produced and pinpoint it. The analysis of a sequence through BLASTn [16] is also facilitated, the download of the alignment and the FASTA files of every or some representative sequences, multiple graphs and a customizable report to allow the user to select the information he or she may deem as relevant.

Key words

New generation sequencing: Collection of methods and techniques that allow a high-throughput capability for DNA sequencing.

CRISPR-CAS9 : Clustered Regularly Interspaced Short Palindromic Repeats asscociated protein 9. Union of nucleotide sequence with a protein capable of identifying specific DNA regions and perform cuts on its double chain.

Genetic mosaicism: Genetical condition that describes an individual with multiple differing genotypes coexisting in different cells or tissues.

Variant calling: Process by which mutations on samples are identified.

Pathogenic variant: Mutation that translates into a decrease or loss of function of a necessary gene.

***paired-end*:** DNA sequencing method where both DNA strands are read in parallel, giving rise to 2 FASTQ files for the same sequence.

Agradecimientos

Este proyecto no habría salido adelante de no ser por la confianza depositada en mi tanto por Miguel Angel Moreno como por Val Fernández. Especial mención a Matias Morín que a pesar de no aparecer como tutor echó horas extra para enseñarme y ayudarme.

Pero ante todo, se lo agradezco a mi familia, por la comprensión y paciencia que han mostrado durante todo este tiempo. A mis compañeros por mantenerme cuerdo. Y a mi pareja, por mantenerme feliz.

Índice general

Índice de Figuras

Índice de Tablas

1 Introducción

1.1. Estructura del documento

Comenzaremos este documento con la motivación del mismo, los objetivos y la metodología. Durante la metodología y plan de trabajo comentaremos los distintos puntos de interes, su intención a la hora de proponerlos además de los pasos que se seguirían para intentar llevar a cabo estos objetivos.

Proseguiremos haciendo una breve descripción y analisis del Estado de Arte en el desarrollo de webs de análisis de datos de NGS. Específicamente en aquellas centradas en datos provenientes de experimentos de edición génica mediente el uso de la tecnología CRISPR-CAS9 .

Tras ello comenzaremos con el desarrollo de la aplicación, comentando el proceso de ejecución de los objetivos y puntos de interes, además de posibles variaciones a estos que se hubiesen tenido que llevar cabo debido a restricciones o eventualidades.

Tras todo ello se procederá a testear la aplicación mediante el uso de datos reales provenientes de experimentos de edición génica y de pacientes.

Finalmente en la conclusión se llevará a cabo un breve análisis sobre los resultados, se propondrán usos de la herramienta final, además de propuestas para solventar las carencias que hayan podido surgir y para la continuación del desarrollo de la herramienta.

1.2. Motivación del proyecto

Las variantes genéticas de baja frecuencia son importantes en muchas areas del estudio de la genómica. Pueden ser determinantes en el pronóstico y control de la evolución de pacientes con enfermedades raras, en aquellos estudios de investigación que incluyan entre sus objetivos la aplicación de la edición génica basada en la tecnología CRISPR-Cas, y aquellos que generen multiplicidad alelica. [4, 5, 6, 7]

Hasta la fecha se hacía uso de técnicas clasicas basadas en la amplificación y secuenciación por Sanger [17, 18], lo cual presentaba enormes limitaciones, entre las que destacan la imposibilidad de detectar alelos que no tuviesen una fracción alélica mayor del 15-20 %. Gracias al desorrollo de la secuenciación de última generación (NGS) la fuerza de trabajo necesaria para secuenciar

grandes cantidades de muestras se ha reducido en gran medida, y la franja de detección de alelos se desvanece por completo. Pero con ello también se ha generado la necesidad de producir en paralelo herramientas alternativas para el manejo, análisis y comprensión de esta masiva cantidad de datos.

Por tanto, el objetivo general de este TFM consiste en el desarrollo y la implantación de una herramienta bioinformátca que permita la identificación, clasificación y cuantificación de la diversidad alélica finalmente obtenida a partir de los datos crudos obtenidos por NGS. Particularmente se espera detectar todos los alelos minoritarios (aquellos que representan menos del 10 % de la fracción alélica).

1.3. Objetivos y enfoque

Los objetivos concretos pueden resumirse en estos tres puntos:

- **1.** La caracterización cuantitativa y discreta de fenómenos de mosaicismos somáticos asociados a los fenómenos de terapia génica natural y en la edición genética por CRISPR-CAS9 .
- **2.** La identificación de multiplicidad alélica generada tras la edición genética por CRISPR-CAS9 .
- **3.** La cuantificación de una forma fiable y reproducible del porcentaje de edición génica en el locus de interés (on-targets) en muestras de pacientes y el análisis de edición génica en off-targets, que es esencial para la traslación de las aproximaciones terapéuticas del sistema CRISPR-CAS9 a la clínica.

1.4. Metodología y plan de trabajo

El proyecto se desarrollará principalmente en el entorno de programación R [10]. El programa de análisis tiene como objetivo ser ejecutado en un servidor para permitir su acceso via web, de manera que se pretende hacer uso del entorno ShinyR [11]. Se testeará la aplicación usando datos (reads en .fastq) de secuenciación masiva provenientes de experimentos de NGS realizados en humanos o modelos animales proporcionados por el departamento de genética del Hospital Ramón y Cajal.

1.4.1. Metodología

El diseño en esencia se subdividirá en varios módulos:

- **Módulo 1** Análisis de las secuencias (reads). Constará de dos tareas:
  - · **T1.1** - Trimming de las reads mediante eliminación de adaptadores e index necesarios para el proceso de secuenciación de Illumina.
  - · **T1.2** - Generación de secuencias contig mediante ensamblado de las reads que provienen de la secuenciación de la misma molécula (forward y reverse), aproximación *paired-end* de Illumina, para conseguir una secuencia contig que represente un alelo independiente.

- **Módulo 2** Clustering de las secuencias.
  - · **T2.1** - Alineamiento de los contigs con una secuencia de referencia o con una secuencia *target* particular.
  - · **T2.2** - Agrupación de los alelos idénticos identificados para generar las distintas clases y estimar su frecuencia

- **Módulo 3** Visualización del análisis de resultados via web.
  - · **T3.1** - Visualización del número y frecuencia de las clases alélicas identificadas.
  - · **T3.2** - Visualización del alineamiento a nivel de secuencia de los distintos alelos identificados mostrando las diferencias en cada posición con respecto a la secuencia de referencia utilizada.
  - · **T3.3** - Visualización de estadísticas de variabilidad por posición de la secuencia analizada en base a las distintas clases alélicas obtenidas.

- **Módulo 4** Generación informe de resultados.
  - · **T4.1** - Generación automática de un archivo PDF de resultados que integre los análisis realizados.

A continuación se muestra una tabla resumiendo lo expuesto arriba con las horas estimadas asociadas.

Tareas / subtareas	**Horas**
T1. <**Módulo 1**: Análisis de las secuencias (reads)>	
T1.1 Trimming de las reads	50
T1.2 Generación de secuencias contig	50
T2. <**Módulo 2**: Clustering de las secuencias >	
T2.1 Alineamiento de los contigs mediante pairwise sequence alignment	50
T2.2 Agrupación de los alelos idénticos identificados	50
T3. <**Módulo 3**: Visualización del análisis de resultados via web >	
T3.1 Gráfico para la visualización de frecuencia de las clases alélicas identificadas	25
T3.2 Visualización del alineamiento a nivel de secuencia	25
T3.3 Visualización de estadísticas de in-dels y Gráfico de barras indicando el porcentaje de variabilidad	25
T4. <**Módulo 4**: Generación informe de resultados >	
T4.1 Generación automática de un archivo PDF	25
Total Horas	300

Cuadro 1.1: **Metodología**

1.4.2. Plan de Trabajo

Partiendo de un cupo máximo de 300 horas, se estima que las primeras 100 sean dedicadas integramentes al módulo 1, dividiendo el tiempo equitativamente entre el trimming de las read y la generación de secuencias contig. Este módulo se llevará acabo en las primeras tres semanas.

El módulo 2, clusterización de las secuencias, se estima que requerirá las siguientes 100 horas. El objetivo es también llevarlo a cabo en las siguientes tres semanas.

Las restantes 100 serán dedicadas a los módulos 3 y 4. Los módulos serán encarados en el orden de propuesta en el diseño, dando prioridad al análisis y clustering de las secuencias. Este módulo se planea llevarlo acabo durante las siguientes tres semanas, permitiendo el tiempo de la última para la revisión del código y de la entrega.

En la siguiente tabla se muestra la planificación de las tareas.

Cronograma semanal									
S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
T1.1	T1.1/T1.2	T1.2	T2.1	T2.1/T2.2	T2.2	T3.1/T3.2	T3.3	T4	Revisión

Cuadro 1.2: **Plan de Trabajo**

2

Estado del arte

2.1. Introducción

Hoy en día, el procesado de archivos de secuenciación masiva está considerablemente desarrollado. Existen múltiples protocolos para manejar los archivos fastq y procesarlos hasta obtener un archivo depurado capaz de ser interpretado, bien para la obtención de variantes puntuales (SNV), de numero de copia (CNV), de edición génica o de multiplicidad alélica. Sin embargo el desarrollo de CRISPR-CAS9 y los datos generados por su uso suponen cierta problemática.

2.2. Nacimiento y evolución de CRISPR-CAS9

Las repeticiones palindrómicas cortas agrupadas y regularmente interespaciadas (CRISPR, por sus siglas en inglés), son una familia de secuencias encontradas en organismos procarióticos que tienen un rol activo a la hora de componer el sistema defensivo de las bacterias y las arqueas contra virus invasores. Las secuencias CRISPR derivan directamente de virus que anteriormente habían invadido el organismo. De esta manera sirven como referencia a la hora de detectar subsiguientes invasiones de virus similares. [19]

Tras la detección de una invasión vírica es cuando la enzima CRISPR-asociado9 (CAS9), siguiendo las secuencias CRISPR como guía, corta las secciones específicas de la doble cadena de DNA complementaria a CRISPR. De esta forma destruye al intruso, y únicamente al intruso por la expecificidad de CRISPR y asimila su secuencia para enfrentarse a nuevas invasiones.

En organismos más complejos que un virus, el complejo CRISPR-CAS9 no tiene un efecto tan destructor. Ocurre que al seccionar la doble cadena de DNA entran en acción mecanismos de reparación de DNA. Existen dos mecanismos principales que llevan a cabo tareas de reparación; Reparación dirigida por homología (HDR) y la unión de extremos no homólogos (NHEJ) [20]. El proceso de HDR hace uso de una secuencia molde para reparar la cadena de DNA, copiando esta secuencia en el lugar donde el daño había sido causado. El proceso de NHEJ sin embargo carece de molde para llevar a cabo su reparación, de modo que la unión de los extremos seccionados no tiene por qué ser igual que antes del daño, introduciendo así cambios de base e in-dels impredecibles en la cadena original.

El proceso de HDR junto con los precisos cortes de CRISPR-CAS9 es lo que se explota para llevar a cabo con éxito la edición génica. Al inyectar el complejo CRISPR-CAS9 junto a una secuencia que haga de molde y que contenga la alteración que se desea introducir, se puede aprovechar el mecanismo de HDR para que el organismo objetivo introduzca esa alteración deseada. Sin embargo como todo proceso teórico, a la hora de llevarlo a la práctica surgen problemas, y es que a pesar de disponer un molde, los procesos de HDR y NHEJ funcionan de forma paralela, introduciendo no solo la mutación deseada si no tambien variantes puntuales off-target incontrolables y por lo tanto producen un efecto de mosaicismo genético.

Combrobar que la mutación por HDR ha sido introducida es necesario, pero el control de las NHEJ puede que todavía lo sea más. Estas mutaciones al fin y al cabo puede que hayan llevado a cabo cambios patogénicos o puede que no. Existe la posibilidad de que en una misma secuencia se hayan expresado ambos metodos de reparación y obtengamos un alelo mixto (HDR-NHEJ). La patogenicidad de la mutación por NHEJ es entonces clave para determinar si el experimento continua o no. Puede que se encuentre una variante que se clasifique como NHEJ pero que sin embargo sea muy frecuente o esté bien apartada de la zona de acción de CRISPR-CAS9 , dando a entender que el proceso de NHEJ no sea el origen de tal variante. Por estos motivos es necesaria la visualización clara a nivel de secuencia de estas mutaciones.

2.3. Herramientas

En la actualidad están surgiendo múltiples herramientas diseñadas para el análisis y la visualización de datos NGS. Concretamente para experimentos de CRISPR-CAS9 destacan por su simplicidad y potencia CRISPResso [2] y CRISPR-GA [3]. Ambas herramientas procesan e identifican mutagénesis en unos datos de secuenciación proporcionados, sin embargo la información devuelta al usuario se limita a resúmenes de variantes agregadas, como se muestran en las Figuras 2.1 y 2.2. De manera que para ciertos estudios, como para la cuantificación de mosaicismos y de la edición específica de un alelo, estas herramientas resultan inapropiadas.

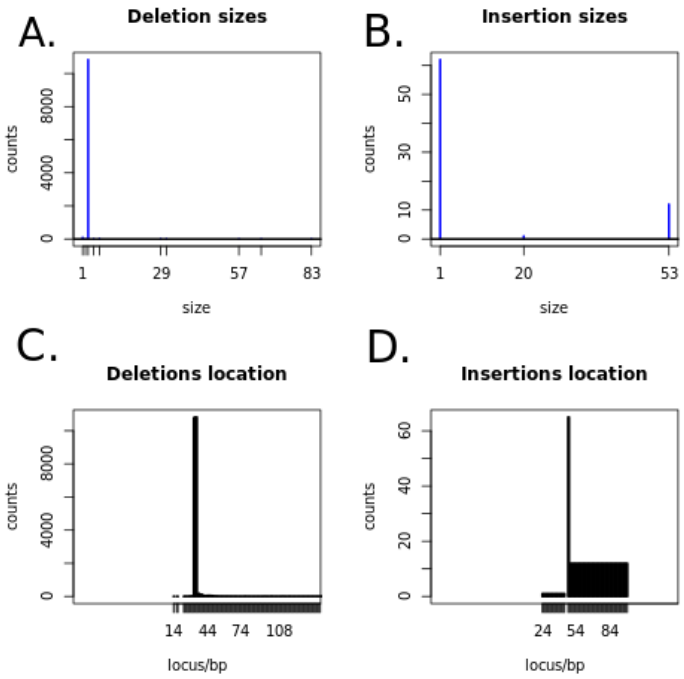


Figura 2.1: **Resultados tras análisis por CRISPR-GA. A) Distribución del tamaño de deleciones B) Distribución de tamaños de inserciones. C) Distribución de deleciones por posición D) Distribución de inserciones por posición**

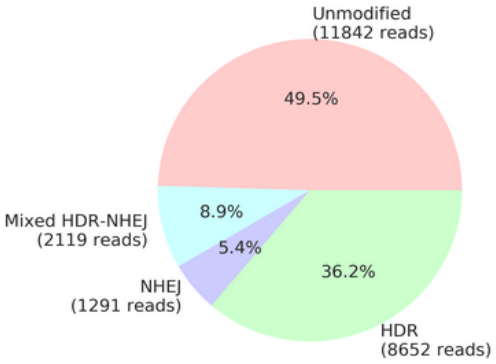


Figura 2.2: **Gráfico de tarta representado por CRISPRESSO. Cuantificación de la frecuencia de edición determinado por el porcentage y número de lecturas de secuencias mostrando alelos modificados e inmodificados. Los alelos modificados quedan subdivididos en NHEJ, HDR y mezcla de HDR-NHEJ.**

3
Desarrollo

Los módulos 1 y 2, análisis de las secuencias y clustering, fueron desarrollados de forma independiente a los módulos 3 y 4. Tanto la visualización del análisis de resultados via WEB como la generación del informe de resultados fueron llevados a cabo en el entorno R de shiny [11] haciendo uso de los paquetes tidyverse [21] para el manejo de datos, plotly [22] para su visualización y rmarkdown [23] para el informe.

Sin embargo para los procesos de análisis y el clustering de las secuencias se usaron paquetes envueltos en Perl. La elección de usar Perl sobre Python, a pesar de que el segundo es más sencillo de programar y de entender, es debido a la mayor eficiencia en la manipulación de cadenas de carácteres de Perl ya que este coge prestados los comandos de sed y awk de LinuxOS. Debido a que vamos a trabajar con ficheros de gran volumen sobre los que es posible que queramos manipular las entradas directamente, se decidió mantenerse en Perl.

El análisis de las secuencias y el proceso de clustering se encuentra esquemáticamente resumido en la figura 3.1 en la página 13.

3.1. Analisis de las secuencias

Para el análisis de secuencias se requiere la introducción manual de los datos de NGS a analizar en formato FASTQ o GZ (comprimido), además de un archivo FASTA con la secuencia de referencia, normalmente aquella considerada *wild type*, siendo *wild type* la secuencia que se encuentra en la naturaleza sin haber sufrido mutación o edición por CRISPR-CAS9 . La introducción de una secuencia objetivo (Target) es opcional pero altamente recomendada. Esta secuencia sería aquella en la que se ha pretendido modificar la secuencia de referencia, en el caso de haber usado CRISPR-CAS9 , o que contiene una mutación que se pudiera estar buscando.

3.1.1. Herramientas

Para llevar a cabo el análisis de las secuencias se hace un uso concatenado (pipeline) de varias herramientas con diferentes propósitos:

• **cutadapt[24]:** *Trimming* de los adaptadores de secuenciadores.

- **biostrings [25]:** Gestor de secuencias DNA y RNA en R.
- **prinseq[26]:** *Trimming* de bases por longitud fija, filtrado de las secuencias por calidad y longitud mínima.
- **ea-utils[27]:** Fusiona dos lecturas de *paired-end* en los extremos solapantes.
- **seqtk[28]:** Tranforma los archivos FASTQ en archivos FASTA filtrados.

3.1.2. Aplicación

Trimming

El comienzo del pipeline de análisis viene dado por la sección de *trimming* . Primero hay una comprobación de si se han aportado parámetros para esta operación, ya que el *trimming* es opcional. Se crearon dos opciones de *trimming* por separado, el *trimming* de residuales y el *trimming* de primers. El *trimming* se lleva a cabo tanto en nuestras lecturas como en los archivos FASTA de referencia y target, en caso de que estas contengan también los primers.

Comenzamos con el *trimming* de los adaptadores . Los secuenciadores de Illumina anexan a cada extremo de las secuencias de las muestras otra pequeña secuencia cuyo fin es que se adhiera a la célula de flujo [29] donde comenzará el proceso de replicación. El propio secuenciador tras finalizar su análisis retira las secuencias de sus adaptadores del archivo FASTQ, tras lo cual quedan unicamente las secuencias de las muestras y los primers utilizados, si es que ha sido un experimento llevado a cabo por amplicones [30] . Sin embargo la realidad es que los secuenciadores en múltiples ocasiones no retiran por completo los adaptadores , dando lugar a secuencias con muy dispares longitudes.

Es por eso que es necesario un filtrado inicial de adaptadores , el cual se lleva a cabo con la herramienta cutadapt. Esta, al introducirle las secuencias de los adaptadores , hará una búsqueda de esa secuencia a lo largo de cada lectura hasta encontrar una secuencia de bases de suficiente similitud. Se permite una tolerancia del 10 % en las muestras, de modo que aunque exista cierta variabilidad entre los adaptadores, ya sea por un cambio de base, inserción o deleción, este sería encontrado y retirado igualmente. El comando sería el siguiente:

```
> cutadapt -a $Adapter_R1;max_error_rate=0.10 -A $Adapter_R2;
    max_error_rate=0,10
```

Se prosigue de nuevo con otro proceso de *trimming* pero esta vez de primers. Debido a que el grueso de las muestras con las que se va a trabajar van a provenir de procesos de amplificación por amplicones, debemos de ser capaces de indentificarlos y retirarlos de nuestras secuencias. En especial por que estos acumulan numerosos cambios de base e indels (inserción-deleción). Todos estos errores serían detectados posteriormente como posibles mutaciones. Concretamente un 30 % de las mutaciones detectadas sin haber retirado los primers desaparecen una vez estos se retiran.

Debido a que el diseño de los primers es dependiente del experimento no se puede usar una secuencia que valga para todos los casos, como ocurre con los adaptadores. Se debe introducir la secuencia bien manualmente o mediante un archivo FASTA. Si se hace a través del archivo FASTA se asumirá que las secuencias proporcionadas pertenecen ambos al archivo Forward, en caso de estar llevando a cabo un análisis *paired-end* . Para extraer las secuencias de los primers del archivo Reverse se calculará la reversa complementaria mediante la función *reverseComplement()* del paquete Biostrings y se mostrará al usuario para que este confirme su correcto cálculo. Tras ello se procederá al *trimming* :

```
> cutadapt -a $Primer_R1;max_error_rate=0.10 -A $Primer_R2;
    max_error_rate=0.10
```

Se introduce también una tolerancia de un 10 % debido a esos errores que aparecen en los primers.

El *trimming* por tamaño es también posible, tanto para los adaptadores como para los primers, sin embargo no es aconsejable. Esto es debido a que los adaptadores a veces sí que han sido retirados por el algoritmo del secuenciador, de manera que si se recorta uniformemente cada lectura estarás recortando en muchas secuencias la sección del primer, produciendo un filtrado inadecuado. Además, en todos aquellos casos en los que hay un adaptador residual y un primer, siempre es posible que estos contengan variaciones que no sean de cambio de base, si no de indel, modificando el tamaño de las secuencias y provocando o bien un residuo o un *trimming* de la secuencia que se intenta filtrar.

La opción se mantiene por la posibilidad de que el usuario no conozca las secuencias de los primers o los adaptadores usados pero quiera aun así retirarlos en la medida de lo posible a pesar de las contra-indicaciones.

Filtrado de calidad y fusión

Se prosigue con un filtrado de calidad y tamaño llevado a cabo mediante prinseq. El motivo por el que se lleva a cabo este tipo de filtrado es debido a que la posibilidad siempre existe de que el secuenciador haya cometido un error al determinar una base como la que es, de modo que en los archivos FASTQ se incluyen informes de calidad sobre todas y cada una de las bases. El algoritmo utilizado por illumina para determinar las calidades de sus propias lecturas no es de código abierto, pero la escala utilizada para interpretarlo es la ya standard Phred33 [31]. Mediante prinseq, se retiran todas aquellas lecturas que no superen un Q=20 (Probabilidad de base erronea = 0,01) además de las que no cumplan una longitud mínima introducida por el usuario. El comando utilizado sería equivalente al siguiente:

```
> prinseq-lite.pl -fastq R1.fastq -fastq2 R2.fastq -min_len $minLen
    -min_qual_mean 20 -output_good good.fastq -output_bad bad
```

Tras haber filtrado correctamente todas nuestras lecturas pasamos a fusionar nuestros archivos en caso de estar trabajando con datos de *paired-end* . Mediante el uso de ea-utils comprobamos cada lectura de cada archivo identificando su homóloga, proporcionándole unos valores de mínimo solapamiento (Nm) y máxima diferencia (Np), estos determinan la disparidad entre dos lecturas homólogas, de este modo descartándolas ambas o fusionandolas. De base estos valores quedan establecidos como Nm = 8 % y Np = 6 %. Este proceso no es necesario en los casos en los que los datos sean de tipo single-end.

```
> ea-utils/clipper/fastq-join -p $Np -m $Nm --output joined
    --input good_1.fastq good_2.fastq
```

Tras este pipeline quedarían los datos listos para el proceso de clusterización.

3.2. Clustering de las secuencias

Durante el proceso de clustering la intención es agrupar todas las secuencias de acuerdo a su similitud entre ellas. De esta manera, aquellas secuencias muy abundantes, como serían las

de un alelo wild-type, se agruparían todas juntas haciéndolas destacar sobre el resto de grupos minoritarios. Los grupos minoritarios deberían de contener aquellas secuencias que muestran mutaciones, que hayan sido alteradas mediante CRISPR-CAS9 o que simplemente hayan acumulado errores de secuenciación que pasasen los filtros de calidad.

Si existen un número concreto de grupos con secuencias distintas entre sí, pero abundantes dentro de sus propios grupos, podemos concluir que nuestra muestra expresa un mosaicismo genético, provocado por CRISPR-CAS9 o por una patología. Es a estos grupos abundantes a los que vamos a prestar la mayor parte de nuestra atención, dado que los grupos minoritarios que representan menos del 0.1 % del total de las secuencias pueden atribuirse a errores de secuenciación. Estos grupos no serán eliminados para que el usuario tenga la posibilidad de observarlas y analizarlas de ser preciso.

3.2.1. Herramientas

Para llevar a cabo la agrupación de las secuencias precisaremos de dos herramientas. SEQTK [28] y CD-HIT [32]

- **SEQTK:** Kit de herramientas para procesamiento de secuencias en formato FASTA/Q.

- **CD-HIT:** Programa de rápida ejecución para la agrupación y comparación de grandes sets de secuencias protéicas o nucleóticas.

3.2.2. Aplicación

Antes de agrupar las secuencias por similitud es necesario su transformación de formato, los archivos FASTQ deben ser traducidos a un formato FASTA que CD-HIT pueda manejar. Para ello utilizamos SEQTK por su versatilidad y simplicidad:

```
> seqtk seq -A joined.fastq > joined.fasta
```

Proseguimos con la agrupación de las secuencias haciendo uso de CD-HIT. Este programa llevará a cabo una comparación secuencia a secuencia, formará grupos representados por la más representativa de cada uno, y continuará alineando el resto de secuencias contra estos grupos.

De encontrar varios alineamientos suficientemente similares, dado ciertos parámetros de cobertura (cov) e identidad (id), las asignará al grupo con mayor similitud. De no encontrar ningún alineamiento que sobrepase los valores mínimos de identidad y cobertura formará un nuevo grupo.

La identidad viene definida como el numero de nucleótidos idénticos en el alineamiento entre el tamaño total de la secuencia más corta y la cobertura como la diferencia en longitud entre las secuencias alineadas. El comando sería el siguiente:

```
> cd-hit-est -c $id -s $cov -i final.fasta -o cluster -g 1
```

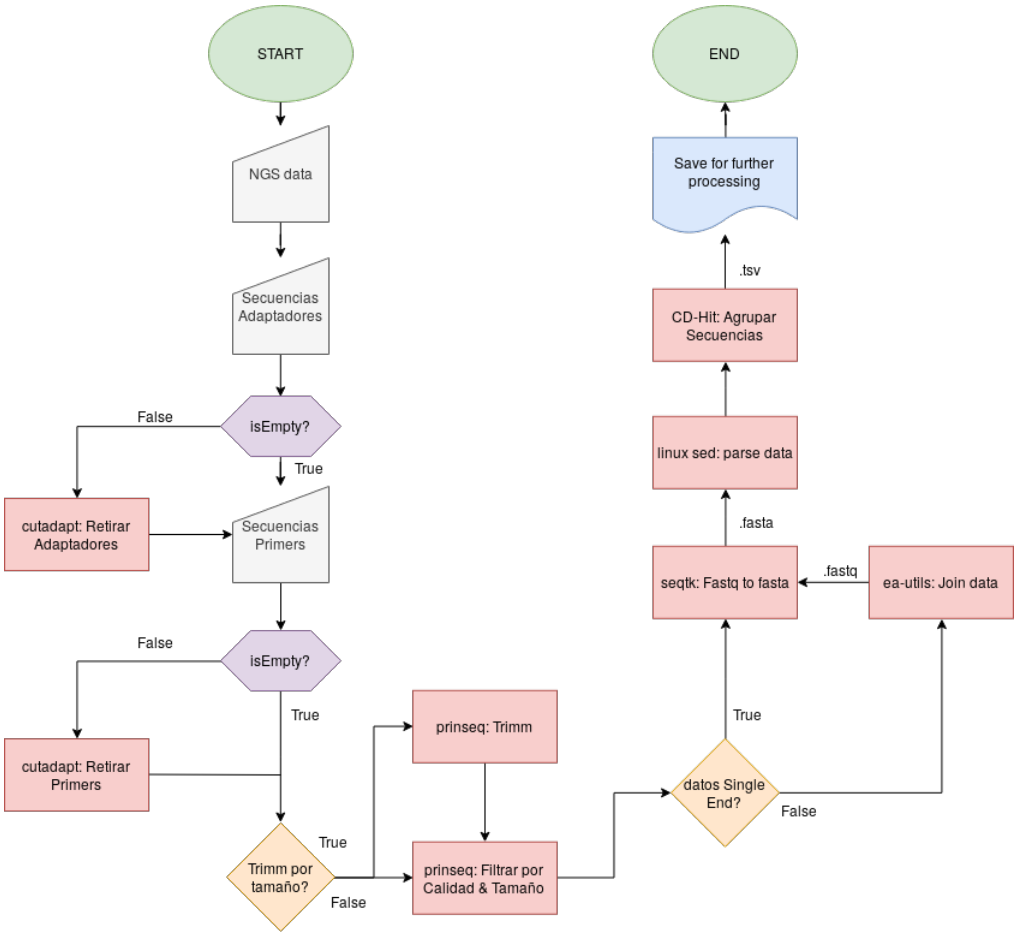


Figura 3.1: **Diagrama de flujo para el analisis y el clustering de las secuencias.**

3.3. Visualización del análisis de resultados via WEB

Una vez llevado a cabo el procesamiento con CD-HIT obtenemos dos archivos que contienen nuestra información de interés. Por un lado un archivo con todos los identificadores y su secuencia asociada en formato FASTA y por otro un archivo con extensión BAK donde se lista cada identificador de secuencia y se le asocia a un grupo y a un nivel de identidad respecto a la secuencia representativa del grupo. De modo que en primer lugar debemos de organizar esta información de forma que sea accesible y manejable a través de R. Para ello transformamos los archivos a un formato de valores separados por tabulaciones (TSV) usando los comandos de sed de Perl.

Tras ello pasamos a usar tidyverse para manejar y concatenar la información en una tabla informativa e interactiva. Todo esto se hará en el entorno de ShinyR para que crear interfaz de usuario que permita la introducción de los datos, la visualización y la interacción a través de un navegador.

3.3.1. Herramientas

- **ShinyR:** Montaje de aplicaciones web interactivas diréctamente desde R.

- **Tidyverse:** Colección de paquetes de R diseñados para las ciencias ómnicas.

Tidyr[33] : Asistente para la ceación y manipulación de datos «ordenados», donde cada columna es una variable, cada fila una observación y cada celda un valor.

Dplyr[34] : Proporciona gramática para la manipulación de datos.

• **Biostrings:** Gestor de secuencias DNA y RNA en R.

3.3.2. Manejo de datos

Comenzamos con el uso del comando sed para formatear los datos proporcionados por CD-HIT. Se tiene en cuenta que este formateo debe de ser capaz de manejar datos provenientes de editores de texto en base UTF8 y de Windows-1252.

El archivo de tipo BAK presenta un formato mostrado en la tabla 3.1:

37	176nt,	>M03698:110:000000000-D2LK5:1:1101:16771:1431...	*
37	176nt,	>M03698:110:000000000-D2LK5:1:1101:16781:1446...	at +/100.00 %
37	176nt,	>M03698:110:000000000-D2LK5:1:1101:14814:1447...	at +/100.00 %
37	176nt,	>M03698:110:000000000-D2LK5:1:1101:14829:1459...	at +/100.00 %

Cuadro 3.1: **Las cuatro primeras entradas del archivo cluster.bak**

Mediante el uso de varios comandos sed en pipes transformamos el archivo en un TSV:

```
> sed 's/at +\\///' cluster.bak | sed 's/\\.\\.\\./\\t/' |
    sed 's/nt, >/\\t/' > cluster.bak.tsv
```

Con la expresión 's/ *string a sustituir* / *sustitución* / ' nos libramos de la información que no nos interesa e introducimos en su lugar indicadores de tabulación (/t).

Por otro lado el archivo FASTA con los identificadores también nos interesa que se encuentre en formato TSV para poder referenciarlo de forma más sencilla con R. Mediante el siguiente código nos aseguramos de que esto se lleva a cabo correctamente independientemente del sistema operativo que genere el archivo FASTA.

```
open(CLS,"cluster");
@fasta = <CLS>; %La variable @fasta almacena cada linea del archivo
                'cluster'
close CLS;

open(CLSout, '>cluster.tsv'); %Creamos el nuevo archivo para ir
                              escribiendo en él linea a linea.

foreach $line (@fasta){
 if($line =~ />/){ %comprobamos la presencia del carácter '>' que
                   indica la presencia del identificador y no de la
                   secuencia
        $line =~ s/>//; %Eliminamos el carácter '>'
        @c = split(/ /,$line); %Subdividimos la línea en dos partes,
                               ya que la parte final del
                               identificador, de existir, sabemos
                               que no nos va a interesar.
        $c[0] =~ s/\s//; %Se elimina todos los caracteres especiales
                         de formato, independientemente de la
```

```
                                codificación.
    print CLSout"$c[0]\t"; %Imprimimos la primera parte de la línea
                                que contendría la parte del identificador
                                que nos interesa.
        } else {
          print CLSout $line; %En caso de que la línea no contenga '>'
                                la línea se imprime tal cual ya que
                                contendrá la secuencia.
        }
}
```

Listing 3.1: **Código de Perl para formateo de FASTA a TSV**

3.4. Creación de Interfaz

El objetivo de la interfaz es que sea intuitiva y completa, pero sin ser abrumadora. Se prioriza la libertad del usuario para observar y determinar la utilidad de los resultados, de manera que se toma la decisión de no eliminar nada, si no ocultar de forma informada y así ni saturar ni asumir qué información no es relevante.

La interfaz consiste de dos menús principales. El primero, la selección de la herramienta a utilizar, el Analizador de Mosaicismos u otra que se pueda querer añadir a más adelante.

El segundo menú se centra en la visualización de los resultados, lo componen tres pestañas; Clustering & Alignment, Graphics y Download. Cada uno desplegará un set de opciones de visualización y manejo de los datos específicos.

Sin embargo, antes de visualizar, el usuario debe de ser capaz de introducir los datos de experimentos de forma simple e intuitiva. De modo que se decidio hacer un diseño columnar por bloques que guien al usuario en el proceso de efectuar su análisis. Esta columna de cuatro partes se puede observar por partes en la Figura 3.2, subfiguras A), B), C) y D).

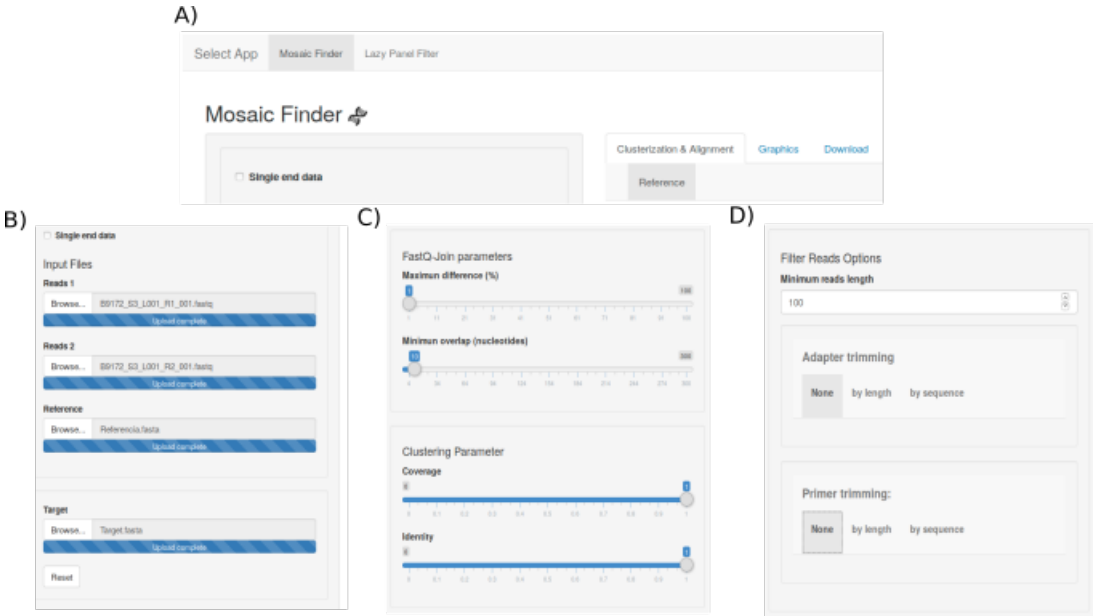


Figura 3.2: **A) Menús principales B) Introducción de datos C) Selección de parametros de *joining* y *clustering* D) Selección de parámetros de *trimming***

3.4.1. Introducción de datos

Las opciones para la introducción de datos se componen de cuatro partes. La introducción de los archivos, las opciones para el *trimming*, los parámetros para la unión de archivos FASTQ y los parámetros de *clustering*.

Introducción de archivos

La introducción de los archivos tiene 4 entradas, tres obligatorias y una opcional. La opcional sería la entrada de la secuencia Target que contiene la mutación. Es opcional por que el análisis se puede llevar a cabo sin ella, sin embargo al no poder comparar los resultados con el Target, la aplicación no será capaz de encontrar el *cluster* que contiene la secuencia representativa más parecida a ese Target y tendrá que ser el usuario quien tenga que llevar a cabo esa búsqueda observando cada entrada particular. Las otras tres entradas son obligatorias y si no se han introducido la aplicación no se ejecutará anticipando que en el proceso vamos a toparnos con un error. La entrada 1 y 2 corresponden a los archivos FASTQ crudos sobre los que se llevará a cabo el análisis descrito en la sección de Análisis 3.1. Cada entrada corresponden al *Forward* y *Reverse* reads respectivamente. Se aceptan tanto archivos FASTQ como archivos comprimidos, detectado de forma automática. La última entrada es la correspondiente al archivo de referencia que por lo general indicará cual es la secuencia *wild-type* y será contra la que se llevará el proceso de alineamiento en la sección de Alineamiento 3.4.3. En esta entrada sin embargo siempre podría introducirse otra secuencia como la propia secuencia Target, los resultados serán parecidos pero los valores de mutaciónes e indels se verán acordemente alterados. El motivo es por que el proceso de *clustering* no requiere una secuencia contra la que alinear, y el proceso de alineamiento 3.4.3 tampoco distingue entre una secuencia u otra, asumirá siempre que está trabajando con la secuencia de Referencia.

Estas entradas son específicas de unos datos provenientes de experimentos en *paired-end*, en la Figura 3.2, subfigura A) y B) se puede observar la casilla *Single-end data*. Al marcarla, la entrada para el archivo *Reverse* se ocultará y dejará de ser obligatoria para la ejecución de la aplicación.

Paramentros para *Trimming*

La siguiente sección será la de *Trimming* , esta se subdivide en tres partes. La introducción del parámetro de lectura mínima, el *trimming* de adaptadores y el *trimming* de primers. La primera simplemente indica a la aplicación el tamaño mínimo que debe de tener una secuencia para considerarla viable, esto debe de ser estipulado por el usuario ya que este parámetro será distinto dependiendo del experimento. El *trimming* de adaptadores y de primers se componen a su vez de tres pestañas que determinan el tipo de *trimming* que se llevará a cabo; None, para ningún *trimming* , *by length*, para el *trimming* por tamaño y *by sequence* para el *trimming* por secuencia. La Figura 3.2, subfigura C) se puede observar en su estado por defecto. La pestaña *None* no depliega nada, la pestaña *by length*, despliega una entrada o dos dependiendo de si se analizan datos en *single* o *paired-end* . Estas entradas tan sólo aceptan valores numéricos, ya que hablamos de un *trimming* por número de bases. La pestaña *by sequence* desplegará de nuevo un menú por pestañas que indicará si la secuencia será introducida de forma directa por el usuario (*Text Input*) o si la introducirá a través de un archivo FASTA (*File Input*), por defecto se selecciona la entrada manual de las secuencias y se despliegan dos entradas (o una en *single-end*), además de un visualizador de las secuencias. Las secuencias, tanto en el archivo *Forward* como el Reverse, van flanqueadas por dos secuencias de primers, de manera que tenemos un total de 4 secuencias. A diferencia de los adaptadores que sólo tendrán una por secuencia. Por

ello a esta altura empezamos a tener que diferenciar entre adaptadores y primers. En las Figura C.4 y C.5 en el Anexo C se puede observar el menú de *trimming* desglosado.

En el caso de los adaptadores no hay complejidad, la secuencia introducida bien manualmente o a través de un archivo se desplegará tal cual para que el usuario confirme que se ha introducido sin problema. Sin embargo en el caso de los primers los usuarios al diseñar su experimento suelen tener disponibles únicamente la secuencia de los primers que flanquean, extremo 5' y 3' , de la secuencia *Forward*. De manera que esas son las dos entradas mostradas a pesar de haber otros dos primers en el archivo *Reverse*, sin embargo como estos primers son los reverso-complementarios de los primers de las secuencias *Forward* mediante el uso de la función *reverseComplement( )* de Biostrings obtenemos los primers de la *Reverse*. Se representan las 4 entradas para que el usuario confirme su correcto cálculo como se muestra en la figura .

Paramentros para *Joining*

La sección de introducción de parámetros para el *joining* es específica para los experimentos de *paired-end* , de manera que se oculta para los de *single*. Estos son los parámetros mencionados en 3.1.2 que determinan la máxima diferencia permitida y el solapamiento mínimo a la hora de fusionar dos archivos con lecturas homólogas (complementarias reversas).

Paramentros para *Clustering*

La última sección son los parámetros de *clustering* mencionados en 3.2, por defecto se establecen a 1 para tanto el *id* y como el *cov* (similitud perfecta).

Tras introducir todos los parámetros ejecutamos el pipeline 3.1 que finalizará con el displiegue de una tabla informativa y compacta en la sección de *Clustering Alignment*.

3.4.2. Visualización de Clustering

Despues de llevar a cabo todo el análisis, por defecto se encuentra seleccionado el submenú de *Clustering & Alignment*. La tabla con los *clusters* se mostrará en el lado derecho de la página, como se puede observar en la Figura 4.1 que presenta un pantallazo a pantalla completa. Un máximo de 10 entradas por hoja, ordenadas por el nivel de abundancia, serán mostradas. Cada fila la compone un *cluster* con el identificador de su secuencia representativa, y cada columna la componen distintas variables; la abundancia, la frecuencia, los cambios de base, la longitud de las secuencias, la puntuación del alineamiento, el número de bases que componen la secuencia, la localización a la que se detecta el comienzo y el final de la secuencia de referencia frente a la secuencia representativa, el número de deleciones y el de inserciones.

Si se ha proporcionado una secuencia Target se dará la opción de observar la tabla respecto a la secuencia Target mediante la selección de una pestaña bajo la propia pestaña de *Clustering & Alignment*. Los datos correspondientes serán mostrados justo debajo.

Creación de tabla informativa

Para crear la tabla informativa se parte de los dos archivos producidos por *CD-HIT* que en la sección de 3.3.2 convertimos en formato TSV. Usando *read_ tsv()* de R convertimos nuestros archivos en data frames de R manipulables, *datos$fasta* y *datos$cluster*. Llegado este punto nos aseguramos de que la variable *datos$fasta* no está vacía, en caso contrario se para la aplicación y se devuelve un valor de error indicando que se han filtrado todas las secuencias.

Se precisa mostrar la abundancia de las secuencias en cada cluster, la frequencia relativa al total de secuencias y el identificador de la secuencia representativa. De modo que se suma cada entrada de cada cluster por separado, para la frecuencia de cada grupo se divide su abundancia por el total de secuencias, y se filtran todos los datos por el carácter * que indica cual es la secuencia representativa. El siguiente bloque de código 3.2 recoge estas operaciones llevadas a cabo mediante funciones de *tidyverse*.

```
colnames(datos$cluster) -> c("ClusterN", "Length", "ID", "Identity")

datos$Tabla = datos$cluster %>%
        group_by(ClusterN) %>%
        mutate(Abundance = n()) %>% %sums every repeated entry
        ungroup() %>%
        mutate(Freq = 100 * Abundance / n()) %>%
        filter(Identity == "*") %>%
        select(ID, Abundance, Freq) %>%
        arrange(desc(Abundance))
```

Listing 3.2: **Manejo de los datos de cluster para obtención de valores de Abundancia, Frequencia y secuencias representativas.**

Para obtener los valores de cambios de base e in-dels debemos de llevar a cabo un alineamiento frente a la secuencia de referencia proporcionada, además de frente al target en caso de haber sido proporcionado. El paquete Biostrings de R posibilita estas operaciones mediante su función *pairwiseAlignment()*. Esta función produce una clase especifica denominada *PairwiseAlignments* que contiene un set de alineamientos. Este tipo de clase es muy conveniente por que se le pueden aplicar múltiples funciones numéricas directamente sin necesidad de forzar la clase de antemano en un data.frame [35]. En la dirección CRISPRAL/Modules/Functios.R, en el script Functions.R, en la línea 135 se puede observar la función propuesta para su cálculo.

Tras su ejecución obtenemos dos variables de data.frames, *datos$tmp* y *datos$Tabla*, que comparten una columna que contienen los identificadores de las secuencias de los *clusters*, de modo que con tan solo un *inner_join()* podemos fusionar correctamente ambas tablas.

3.4.3. Visualización de Alignment

Para producir y mostrar cada alineamiento de cada cluster se decidió dar la posibilidad al usuario de seleccionar *clusters* deseados para visualizarlos de uno en uno. Meramente haciendo click con el ratón en la tabla recientemente generada se dará comienzo al proceso de alineamiento según ciertos parámetros. Estos parámetros pueden ser alterados por el usuario ya que serán desplegados nada más seleccionar el *cluster*. Las opciones tienen que ver con el tipo de alineamiento que se desea llevar a cabo. Específicamente, el método de alineamiento y la secuencia contra la que alinear (*Referencia* o *Target* en caso de existir). Los desplegables se pueden observar en la Figura 3.3. Entre los métodos disponibles se encuentran el alineamiento global (algoritmo Needleman-Wunsh [36]), local (algoritmo SmithWaterman [37]), por solapamiento (alinéa de forma parecida a global pero sin tener en cuenta las penalizaciones de los extremos) y glocal (global-local) para situaciones en que ninguno de los dos métodos fuese apropiado [38]. Por defecto se selecciona el alineamiento global, ya que las secuencias con las que se trabajaría habitualmente serían en principio muy parecidas y de tamaños muy similares, ideales para un alineamiente de este tipo.

Renderizado de alineamiento

De modo que dependiendo de qué cluster haya sido escogido se llevará a cabo un alineamiento mediante *pairwiseAlignment()* y se imprimirá en la pantalla. Para la impresión utilizaremos 4 strings de caracteres que desplegaremos juntos. Por un lado, la secuencia de referencia (*ref*), la secuencia representativa del cluster (*query*), el string destacando las diferencias entre ambas (*comp*), y un string que indique la posición de las bases con respecto a la referencia (*pos*).

Los strings de ref y de query no requieren ninguna modificación sin embargo string comp se obtendrá mediante el bloque de código 3.3.

```
comp = character(length( datos$aln2@pattern ))
ref = unlist(strsplit(as.character( datos$aln2@pattern ),
                      split = "" ))
query = unlist(strsplit(as.character( datos$aln2@subject ),
                  split = "" ))

    for (i in 1:length( ref ))
    {
      if (ref[i] != query[i])
      {
        comp[i] = query[i]
      } else {
        comp[i] = "."
      }
    }
```

Listing 3.3: **Obtención del string *comp*, comparación entre ref y query.**

El string con las posiciones se obtiene únicamente a través de la secuencia de referencia, pero teniendo en cuenta las inserciones que se hayan podido introducir en la secuencia query. Este proceso requiere también cierto nivel de complejidad debido a que los números a partir de la decena y luego otra vez de la centena, cambian de tamaño en lo respecto a caracteres. En el script Functions.R en la línea 3 se puede observar la función propuesta para su cálculo y en la Figura 3.3 su despliege final.

3.4.4. Funciones alternativas

Bajo las opciones de alineamiento se muestran el total de lecturas detectadas y en caso de haber proporcionado una secuencia Target, se estima donde se encuentra de acuerdo a la puntuación del alineamiento de todos los *clusters* contra la secuencia Target.

Existe la opción de observar la secuencia del cluster sin que esté alineada, ya que el alineamiento local y a veces el global, puede estimar ciertas secciones de la secuencia como irrelevantes en vez de considerarlos gaps, sobretodo los extremos. De este modo puedes comprobar que el alineamiento está siendo adecuado.

Finalmente, se presentan tres botones con diferentes funciones, el primero permite comprobar la proveniencia de una secuencia que pueda ser enormemente disimilar de la secuencia de *referencia* o *target* mediante la redirección a *nucleotide blast* [16] del Centro Nacional de Información Biotecnológica (NCBI por sus siglas en inglés). Introduce toda la información, a excepción de la base de datos (BD), para que se lleve a cabo la búsqueda de forma rápida y sencilla. La página de *nucleotide blast* permite la introducción automática de la mayoria de su contenido directamente

a través de variables en el propio localizador de recursos uniforme (URL), de manera que según la entrada seleccionada por el usuario se crea un URL específico y se redirige a *nucleotide blast* con él. Se decidió tomar esta vía ya que alojar las BD de NCBI localmente no era posible, debido a limitaciones tecnológicas, y el acceso por vía programática está enormemente limitado.

El desarrollo de la función se encuentra en el script serverMosaic.R entre las líneas 646 y 656.

Los otros dos botones permiten la descarga del alineamiento, archivo PAIR, específico de la función *writePairwiseAlignments( )* de Biostrings, y de un archivo FASTA con la secuencia representativa sin alinear. Los botones de descarga se crean a partir de la función *downloadButtonModule(**function_name, name_shown**)*. Donde ***function_name*** llama a una función definida en el módulo *donwloadFileModule.R* donde se definen varias funciones para el reciclaje de los códigos de descarga de ShinyR, específicos para el tipo de archivo que se quiera proporcionar.

En la Figura 3.3 se puede observar el aspecto final de estas funciones junto al despliegue de un alineamiento ilustrativo.

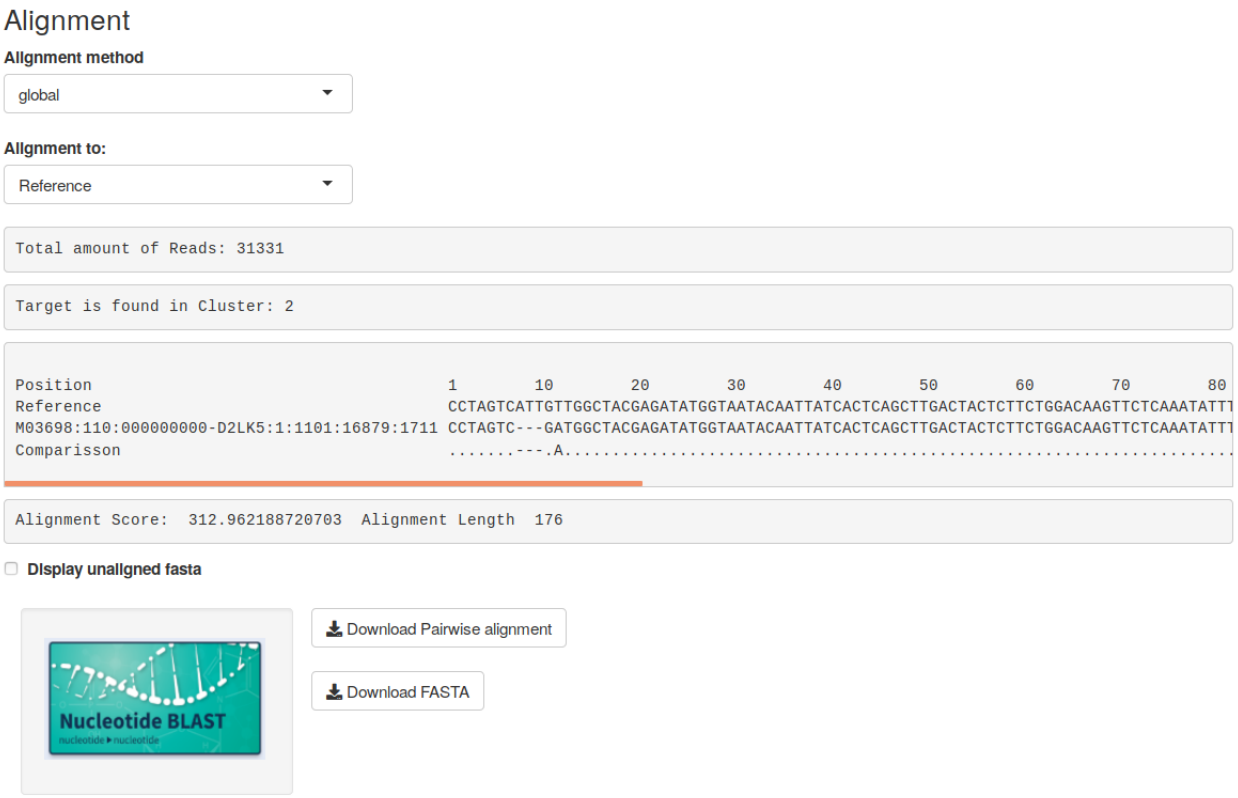


Figura 3.3: **Desplegables para los parámetros de alineamiento, y botones de descargas y análisis por BLASTn.**

3.4.5. Visualización de gráficos

La visualizacion de los gráficos es opcional y por tanto se encuentran en una pestaña aparte. Esta muestra varias casillas marcables que muestran u ocultan los gráficos deseados. En caso de marcarlos todos como viene por defecto aparecerán 6 gráficos que concatenan la información de todos los *clusters* lo máximo posible.

- Gráfico de tarta que muestre los *clusters* mayoritarios por un lado y todos aquellos

minoritarios (<1 %) juntos.

- Frecuencia de cada base por posición
- Distribución de las posiciones de las deleciones por posición
- Distribución de los tamaños de las deleciones
- Distribución de las posiciones de las inserciones por posición
- Distribución de los tamaños de las inserciones

A continuación se describirá el proceso de generación de cada gráfico.

Gráfico de tarta

Para obtener el gráfico de tarta se partió de los datos en data.frame *joinedTabla* obtenidos en la sección 3.4.2. Dado que en este gráfico se precisa únicamente de la frecuencia (Abundancia/Total de secuencias) de cada cluster, se eliminan todas las demás variables de nuestro data frame. Posteriormente se identifican todos los *clusters* cuya frecuencia es menor del 1 % y se almacenarán en una variable propia (*Other*).

El resto de *clusters*, aquellos cuya frecuencia es superior al 1 %, se mantendrán sin alteraciones excepto por su identificador que pasará a ser el número de cluster, ya que este gráfico irá siempre acompañado de la tabla informativa detallada y representar la total extensión del identificador podría estropear su legibilidad en muchos casos.

En el módulo Functions.R, entre las líneas 120 y 135, se puede observar la función desarrollada para eliminar los identificadores y atribuirle a un número variable de *clusters* un identificador acorde.

Frecuencia de cada base por posición

Para continuar con el resto de gráficos se extraen de *joinedTabla* las variables de Abundancia y los identificadores de las secuencias. Ya que queremos extraer posiciones de las bases se obtienen también las propias secuencias a partir de los datos de alineamiento que se lleva a cabo en la sección 3.4.2.

De todas las secuencias representativas se identifica aquella de mayor longitud y se crea un vector de posiciones equivalente al tamaño de esa secuencia. Cada posición es repetida tantas veces como secuencias representativas haya. De la misma forma, nuestro vector con los identificadores de secuencias representativas es copiado y anexado a si mismo tantas veces como posiciones se vayan a representar. Unimos ambos vectores y añadimos otra variable denomida *Character* donde se almacena el carácter de cada secuencia en cada posición. También se tiene en cuenta que cada secuencia representativa representa a un número específico de secuencias, de manera que la abundancia asociada a cada secuencia se almacena también como *Abundancia*. Se prosigue sumando todas las *Abundancias* por cada *Character*, obteniendo así 5 entradas (cada base incluyendo el guión) por cada posición, y un total que representa el número de veces que se observa ese carácter en esa posición. En la tabla 3.2 se puede observar un ejemplo ilustrativo.

Lo siguiente sería normalizar esta tabla y ya tenemos los datos de *frecuencia de base por posición*.

Abundance	Chr	Position
22	A	1
21366	C	1
1	G	1
14	T	1
1	-	2
4	A	2
21383	C	2
7	G	2
8	T	2
1	-	3

Cuadro 3.2: **Ejemplo ilustrativo para ocurrencia de base por posición.**

Distribución de las posiciones de las deleciones por posición

Para obtener la distribución de deleciones por posición simplemente debemos de extraer de la tabla anterior, *frecuencia de base por posición*, aquellas entradas de la variable *Character* que contienen un guión (-) y imprimirlas por su cuenta.

Distribución de los tamaños de las deleciones e inserciones por posición

La obtención de los tamaños de los in-dels requiere volver a los datos proporcionados la función *pairwiseAlignment()*, usando las funciones *nindel()* se extrae del objeto *PairwiseAlignment* los datos referentes a las deleciones y las inserciones. Concretamente por cada alineamiento calcula el número total de deleciones e inserciones, de manera que viene perfecto para el propósito de este gráfico.

Distribución de las posiciones de las inserciones por posición

La obtención de las distribuciones de los inserts requiere del uso de la función *indel()* por que para este gráfico requerimos también del posicionamiento y no sólo de los tamaños de las inserciones. No se podía usar tampoco los datos de *frecuencia base por posición* ya que estos datos contenían tan sólo los datos de las deleciones.

Código

Los códigos desarrollados para estos gráficos se encuentran en el script serverMosaic.R.

- Gráfico de tarta: De las lineas 364 a la 412.
- Frecuencia de bases por posición: De las lineas 364 a la 412.
- Posiciones de las deleciones por posición: De las lineas 459 a la 461.
- Tamaños de las deleciones e inserciones por posición: En el módulo Functions.R de las lineas 61 a la 85.
- Posiciones de las inserciones por posición: De las lineas 463 a la 474.

3.5. Generación de Informe automático

Mediante el uso de RMarkdown se pretende crear un generador de informes automáticos cuyo contenido dependa no sólo de los datos del experimento concreto si no también de lo que el usuario quiera ver en el informe.

El informe se compondrá de dos secciones, los resultados y los gráficos. Los resultados serán representados siempre, sin embargo la opción de gráficos será eliminada en caso de que el usuario no quiera que la contenga.

La sección de resultados se compondrá de la tabla informativa generada en la sección 3.4.2. Sin embargo esta vez el usuario podrá seleccionar varias entradas, de manera que se imprimirán en el informe todas aquellas seleccionadas. En caso de no seleccionar ninguna se imprimirá una cantidad *default* de 10 entradas.

Proseguiremos comprobando si algún gráfico ha sido seleccionado para la impresión y, en caso afirmativo, cuales exactamente. La impresión de gráficos requiere de la transformación de los objetos producidos por plotly a una imágen estática. Esto se llevó a cabo usando la función *export(graph, nombreimagen.pdf)*.

En la Figura 3.4 se puede observar un diagrama de flujo ilustrativo del proceso de generación de informes. El código integro se encuentra en el archivo reports.Rmd.

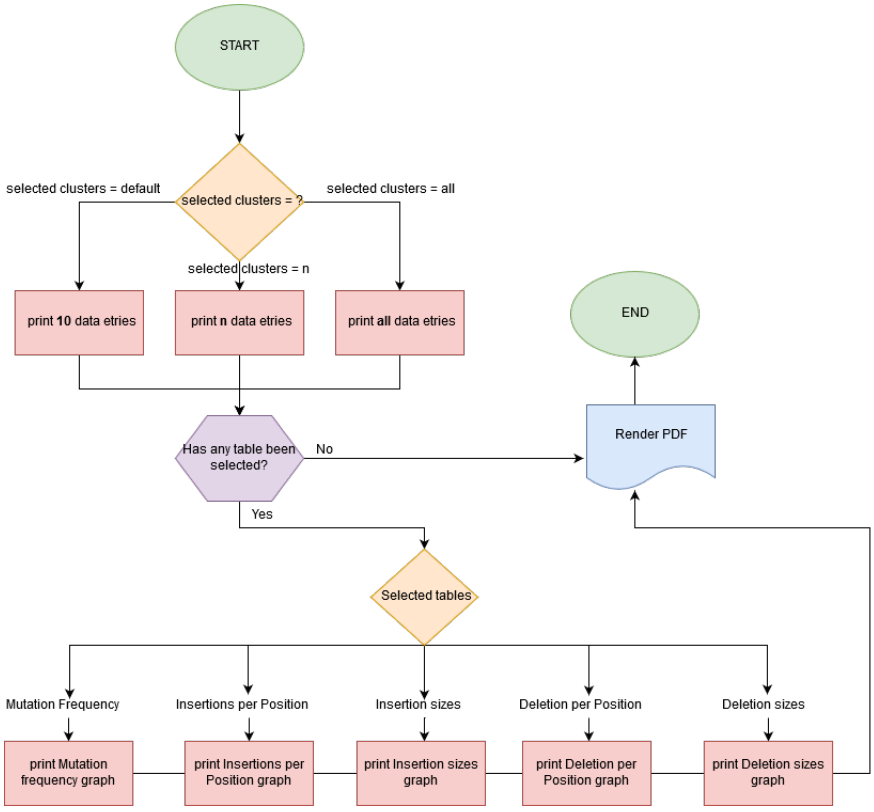


Figura 3.4: **Diagrama de flujo de generación de informes automáticos**

4

Resultados

4.1. Resultado final de la interfaz

En la Figura 4.1 podemos observar el posicionamiento de las entradas de datos a la izquierda, la tabla informativa a la derecha, sobre los alineamientos y los botones de descarga. Podemos observar también la presencia de la pestaña para cambiar de tabla a la del *target*. En la Figura C.1, que haciendo click en la pestaña, la tabla que se despliega presenta resultados muy parecidos a los de la Referencia pero con los datos de las inserciones y las deleciones cambiados como correspondería. La pestaña de gráficos carga adecuadamente como se muestra en la Figura 4.2, y la pestaña de informe permite la selección de múltiples opciones y ajusta sus parámetros acordemente 4.3. Se observa sin embargo que tras generar múltiples informes seguidos se produce un error que impide subsiguientes descargas de informes. La Figura C.2 muestra el error.

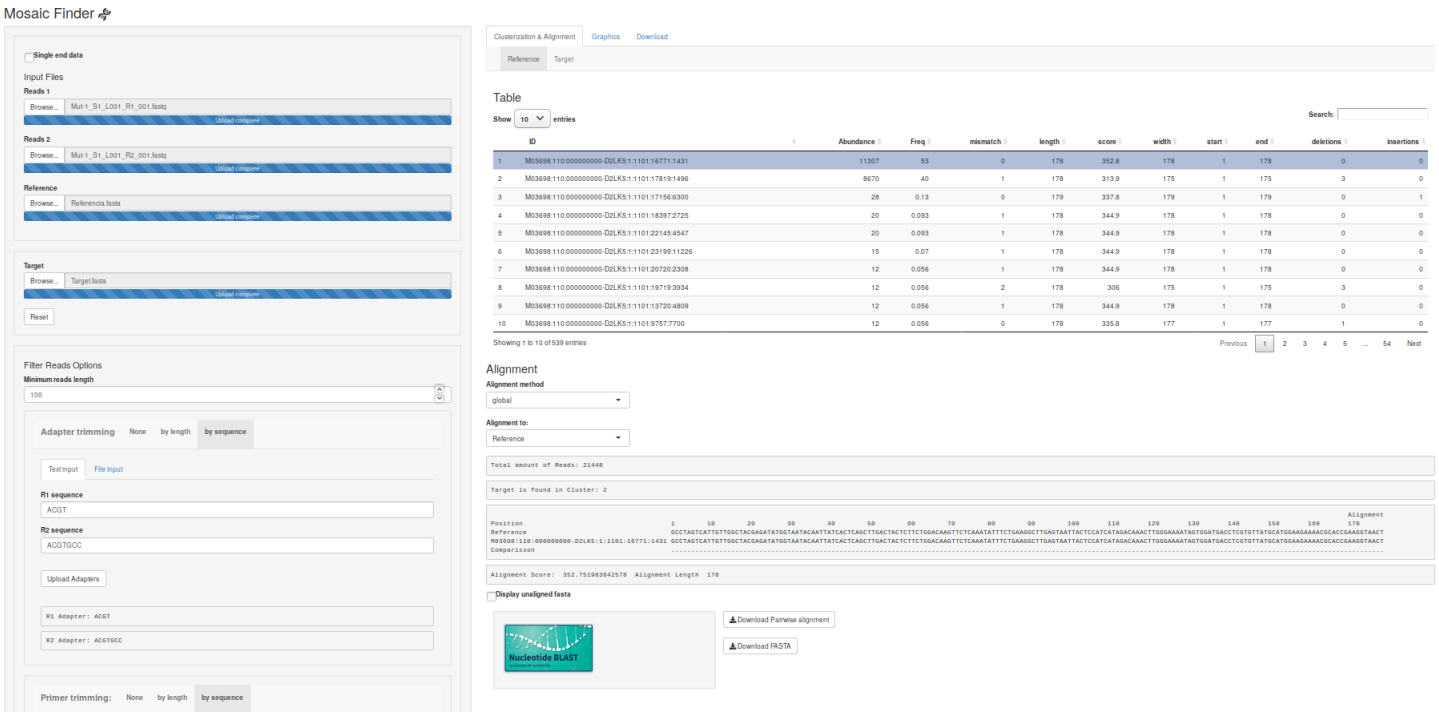


Figura 4.1: **Captura de pantalla de la Interfaz final mostrando parte del menú de introducción de datos a la izquierda, la tabla informativa en su totalidad a la derecha, sobre al alineamiento y botones auxiliares. Datos de ratón mutante.**

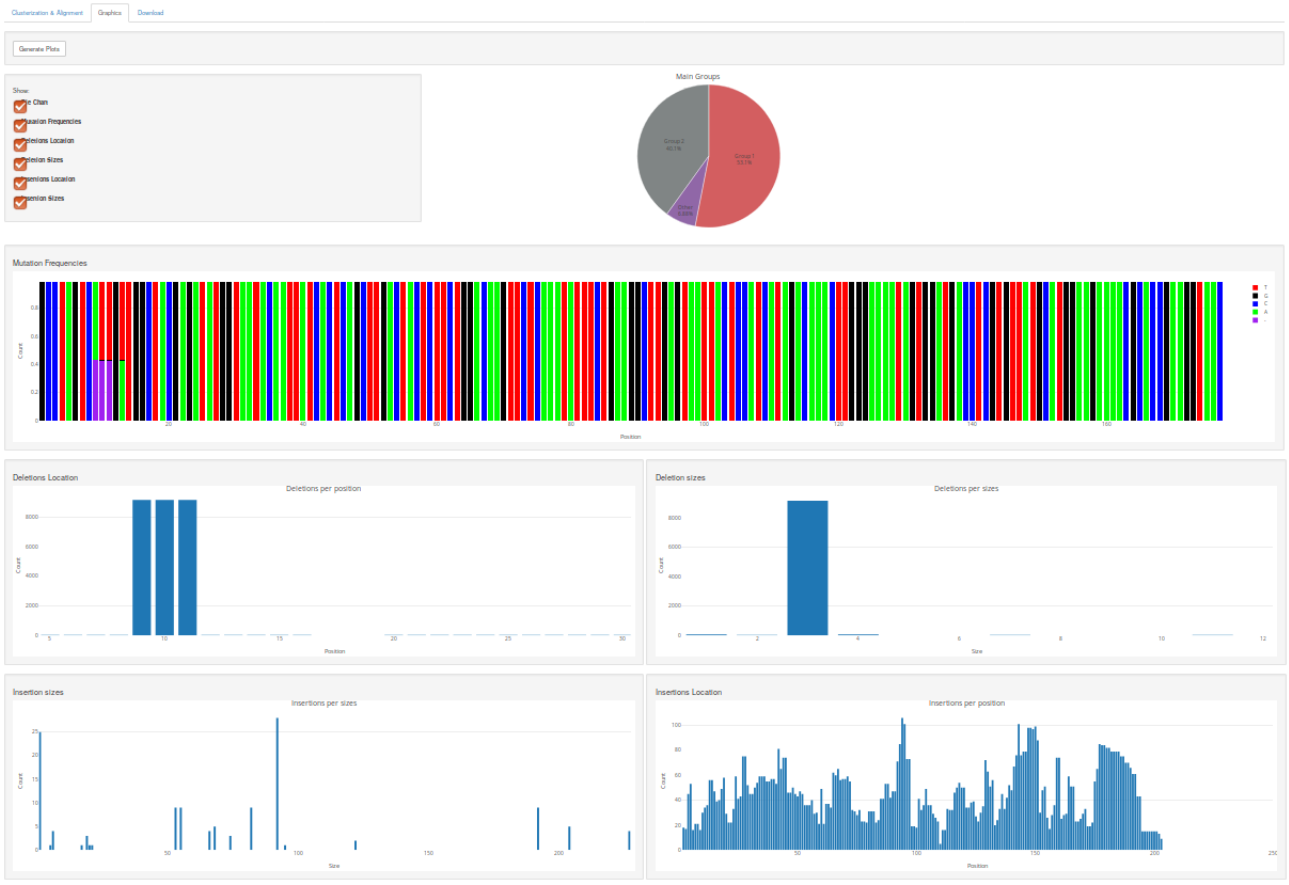


Figura 4.2: **Captura de pantalla de la Interfaz final mostrando todos los posibles gráficos generados.**

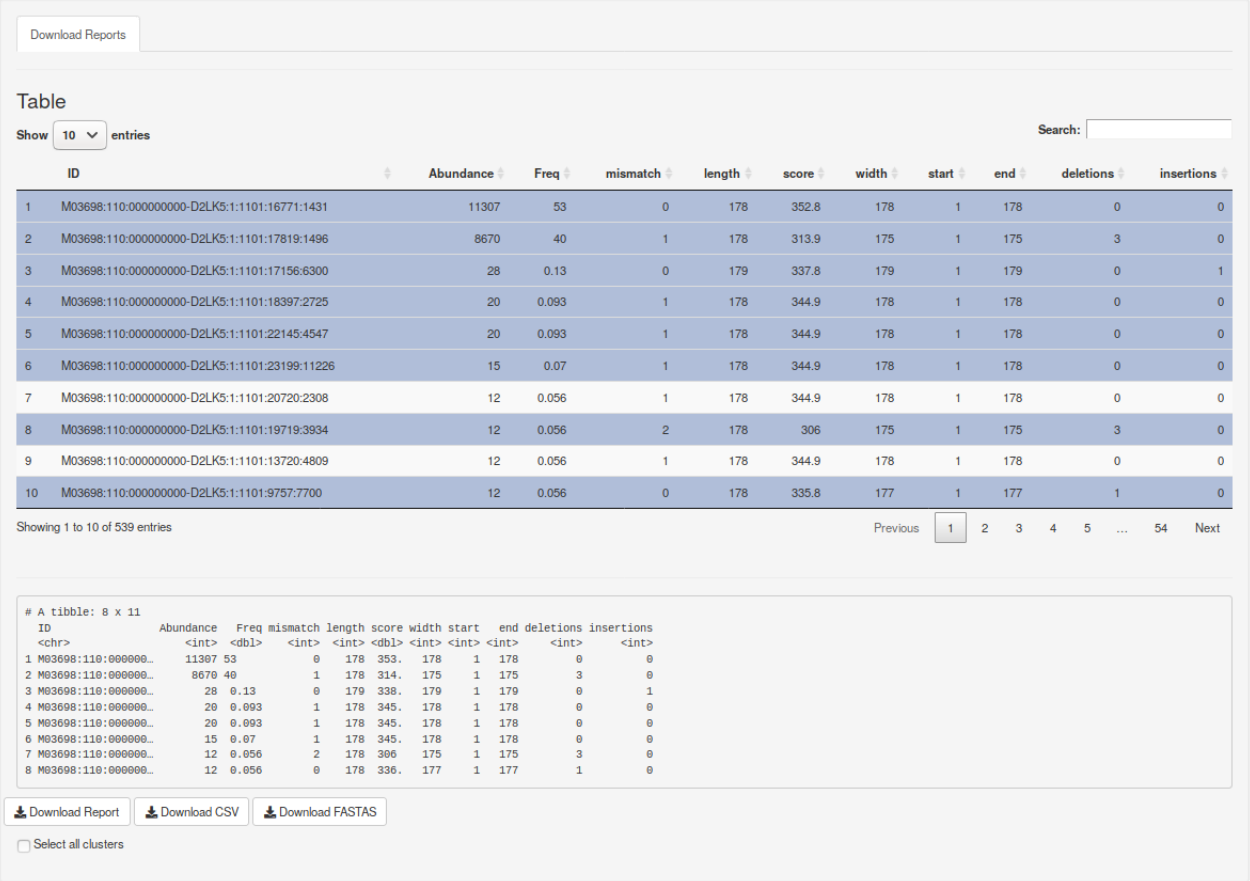


Figura 4.3: **Captura de pantalla de la Interfaz final mostrando la múltiple selección de entradas para el Informe.**

Report

Contents

0.1 Results

Selected clusters: 1, 2, 3, 4, 5, 6, 8, 10 of a total of 539

Table 1: Clusters

Cluster	ID	Abundance	Freq	mismatch	length	score	width	start	end	deletions	insertions
1	M03698:110:000000000-D2LK5:1:1101:16771:1431	11307	53.000	0	178	352.8	178	1	178	0	0
2	M03698:110:000000000-D2LK5:1:1101:17819:1496	8670	40.000	1	178	313.9	175	1	175	3	0
3	M03698:110:000000000-D2LK5:1:1101:17156:6300	28	0.130	0	179	337.8	179	1	179	0	1
4	M03698:110:000000000-D2LK5:1:1101:18397:2725	20	0.093	1	178	344.9	178	1	178	0	0
5	M03698:110:000000000-D2LK5:1:1101:22145:4547	20	0.093	1	178	344.9	178	1	178	0	0
6	M03698:110:000000000-D2LK5:1:1101:23199:11226	15	0.070	1	178	344.9	178	1	178	0	0
8	M03698:110:000000000-D2LK5:1:1101:19719:3934	12	0.056	2	178	306.0	175	1	175	3	0
10	M03698:110:000000000-D2LK5:1:1101:9757:7700	12	0.056	0	178	335.8	177	1	177	1	0

Figura 4.4: **Ejemplo de informe generado.**

4.2. Resultados experimentales

Para continuar comprobando el buen funcionamiento de la aplicación se procede a su examen con datos de experimentos reales. Se usarán dos sets de datos distintos, unos procedentes de un experimento de edición genética en ratones y otro de una línea celular humana. En ambos casos se quería llevar a cabo una mutación en un locus concreto y se esperaba por ello cierto nivel de mosaicismo.

4.2.1. Experimento en ratón

El objetivo de este experimento era obtener unos ratones mutantes en los que a un gen concreto se le delecionan 3 bases y se cambia una. Tras inyectar las células madre de los ratones iniciales, se secuencian y se escoge un fundador que contenga la mutación en suficiente proporción frente al resto de los alelos. Tras seleccionar al fundador se cruza y se obtienen ya unos ratones con un alelo mutado mayoritario.

Análisis del ratón fundador

Se comienza comprobando que el ratón fundador efectivamente contiene la mutación. Introducimos los datos, el archivo con la referencia, el *target*, los adaptadores específicos (MISEQ) y los primers. En la Figura 4.5 se pueden observar los resultados del experimento.

	ID	Abundance	Freq	mismatch	length	score	width	start	end	deletions	insertions
1	M03698:98:000000000-D2NYR:1:1101:15427:1336	4667	31	0	178	352.8	178	1	178	0	0
2	M03698:98:000000000-D2NYR:1:1101:17342:1441	4135	28	0	231	77.8	231	1	231	0	53
3	M03698:98:000000000-D2NYR:1:1101:16964:1477	3893	26	0	244	12.8	244	1	244	0	66
4	M03698:98:000000000-D2NYR:1:1101:20023:2535	552	3.7	7	180	277.6	180	1	180	0	2
5	M03698:98:000000000-D2NYR:1:1101:15836:2330	285	1.9	0	232	72.8	232	1	232	0	54
6	M03698:98:000000000-D2NYR:1:1101:16285:1428	227	1.5	1	178	298.9	175	2	176	3	0
7	M03698:98:000000000-D2NYR:1:1101:5120:7330	12	0.081	1	178	313.9	175	1	175	3	0
8	M03698:98:000000000-D2NYR:1:1101:16170:9612	10	0.067	0	229	87.8	229	1	229	0	51
9	M03698:98:000000000-D2NYR:1:1101:14683:13360	10	0.067	0	244	12.8	244	1	244	0	66
10	M03698:98:000000000-D2NYR:1:1101:11022:4634	9	0.061	1	231	69.9	231	1	231	0	53

Showing 1 to 10 of 635 entries Previous 1 2 3 4 5 … 64 Next

Alignment

Alignment method

global

Alignment to:

Reference

Total amount of Reads: 14869

Target is found in Cluster: 7

```
Position                              1       10        20        30        40        50        60        70        80        90        100       110       120       130       140       15
Reference                             GCCTAGTCATTGTTGGCTACGAGATATGGTAATACAATTATCACTCAGCTTGACTACTCTTCTGGACAAGTTCTCAAATATTTCTGAAGGCTTGAGTAATTACTCCATCATAGACAAACTTGGGAAAATAGTGGATGACCTCGTGTTATGC
M03698:98:000000000-D2NYR:1:1101:5120:7330 GCCTAGTC---GATGGCTACGAGATATGGTAATACAATTATCACTCAGCTTGACTACTCTTCTGGACAAGTTCTCAAATATTTCTGAAGGCTTGAGTAATTACTCCATCATAGACAAACTTGGGAAAATAGTGGATGACCTCGTGTTATGC
Comparisson                           ...........-A.....................................................................................................................
```

Alignment Score: 316.925689697266 Alignment Length 178

Display unaligned fasta

Figura 4.5: **Tabla informativa resultante del análisis de los datos del ratón fundador sobre alineamiento de la entrada 7 identificado como contenedora de la mutación por HDR.**

La aplicación identifica el alelo 7 como el alelo mutado. Hacemos click en él y efectivamente podemos ver en el alineamiento la deleción y el cambio de base. Sin embargo observamos que la entrada del alelo 6 parece contener también la mutación, y además en mayor proporción que el alelo 7 (>1 %). Se procede a comparar ambos resultados e inicialmente, al observar los

alineamientos, no parece que exista diferencia entre ambos alelos. Sin embargo al desplegar las secuencias sin alinear observamos una clara diferencia. El alelo mayoritario que contiene la mutación, el alelo 6, parece tener una base extra al comienzo de la secuencia. Las Figuras 4.6 y 4.7 muestra estos resultados.

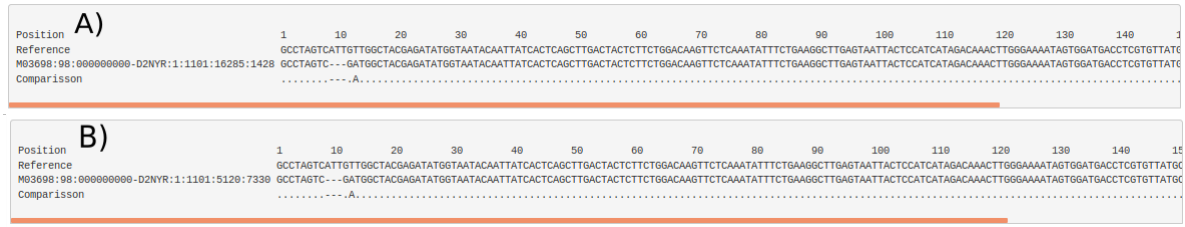


Figura 4.6: **Comparación entre los alineamientos de la entrada 6 A) y 7 B).**

Figura 4.7: **Comparación entre los secuencias sin alinear de las entradas 6 A) y 7 B).**

Esto puede ser debido a; una inserción en la secuencia justo en el extremo que no ha sido identificada por el alineamiento global, que el trimming de la secuencia no se ha efectuado correctamente, o que los primers de esas secuencias contenian una inserción en uno de sus extremos y el software del trimming (*cutadapt*) lo ha considerado ya como parte de la secuencia real. Nos decantamos inicialmente por este último supuesto, dado que los primers tienen una enorme incidencia mutacional. Igualmente, la mutación por HDR se encuentra en la secuencia, con una proporción del 1 %, suficiente para usar este ratón como fundador dado que precisamente los gametos son los que la expresan.

En la sección de gráficos, extrayendo el gráfico de tarta podemos ver el gran grado de mosaicismo que expresa la muestra.

Identificamos varias entradas con un score negativo con respecto a la referencia y se decide analizarlo mediante el módulo de búsqueda de BLASTn. Identificamos la secuencia como proveniente de Homo sapiens, lo que podría indicar cierto grado de contaminación (Figura 4.8).

Figura 4.8: **Resultado de busqueda por BLASTn de una secuencia con un score negativo entre las entradas de la resultante tabla informativa del ratón fundador.**

Análisis del ratón mutante

Tras seleccionar el fundador y obtener un ratón mutante se secuencia y se analiza para buscar la mutación. Los resultados, Figure 4.9 muestran una proporción del alelo mutante mayoritaria,

siendo el segundo alelo más abundante. Al seleccionarlo vemos que efectivamente la mutación es idéntica a aquella del ratón fundador y que no muestra la inserción al comienzo de su secuencia como ocurría en los resultados del ratón fundador.

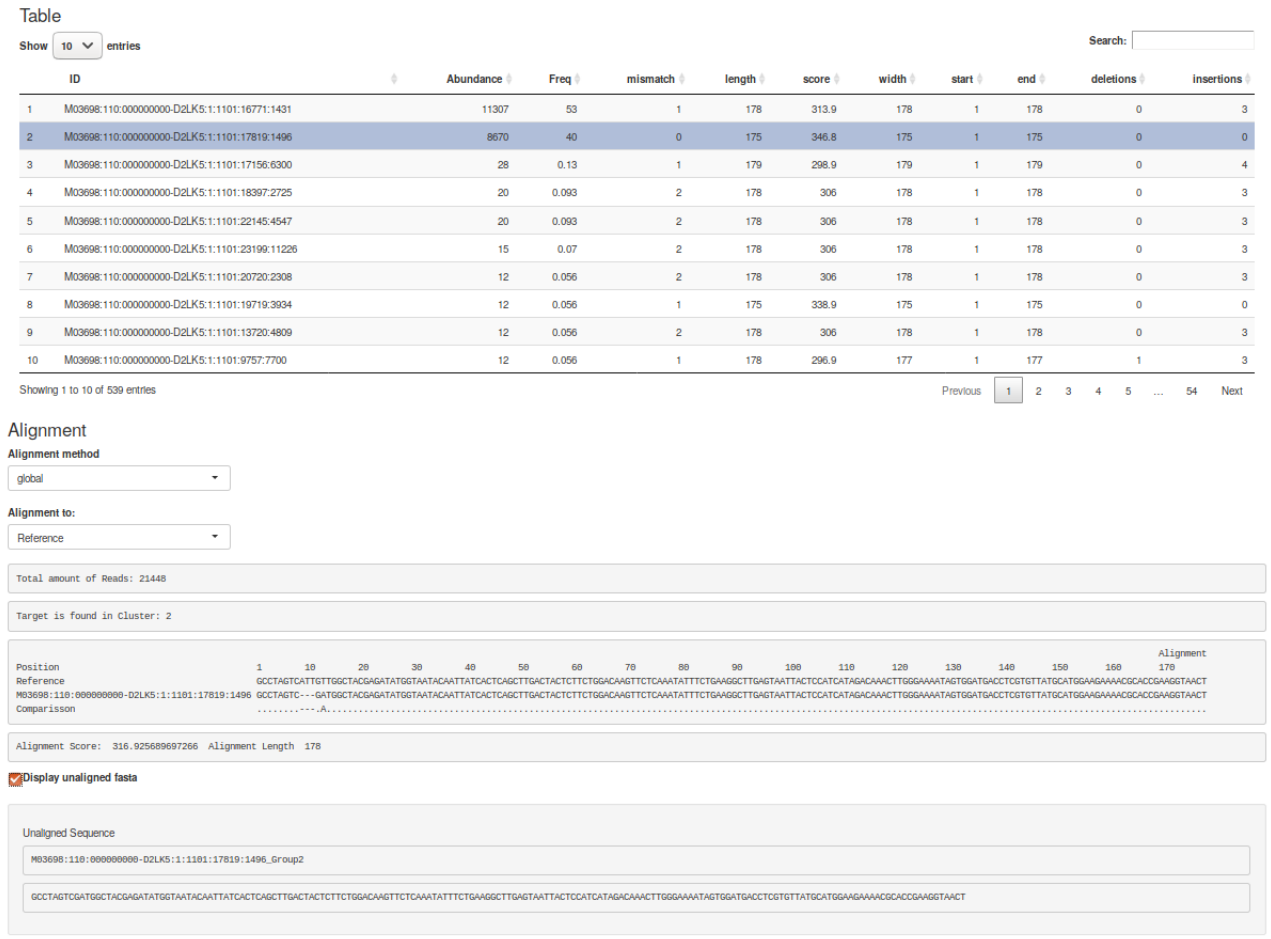


Figura 4.9: **Resultados de análisis ratón mutante.**

El gráfico de tarta muestra los dos alelos mayoritarios de forma clara frente al mayor grado de mosaicismo del ratón fundador (Figura 4.10) y los gráficos de deleciones son coherentes con los resultados de la tabla. Un enorme pico en las posiciones 9, 10 y 11 en el gráfico de *deleciones por posición*, posiciones que se corresponden con la localización objetivo de la mutación, la Figura 4.11 muestra estos resultados.

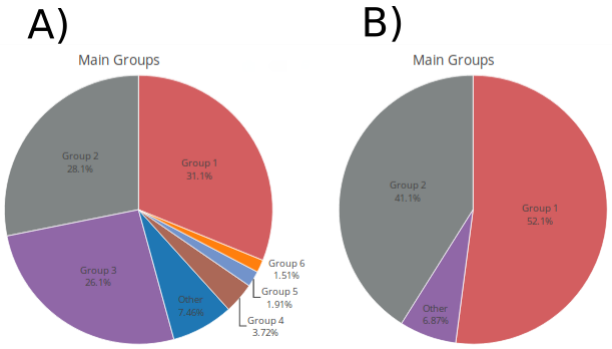


Figura 4.10: **A) Gráfico de tarta tras análisis de datos de ratón fundador. B) Gráfico de tarta tras análisis de datos de ratón mutante.**

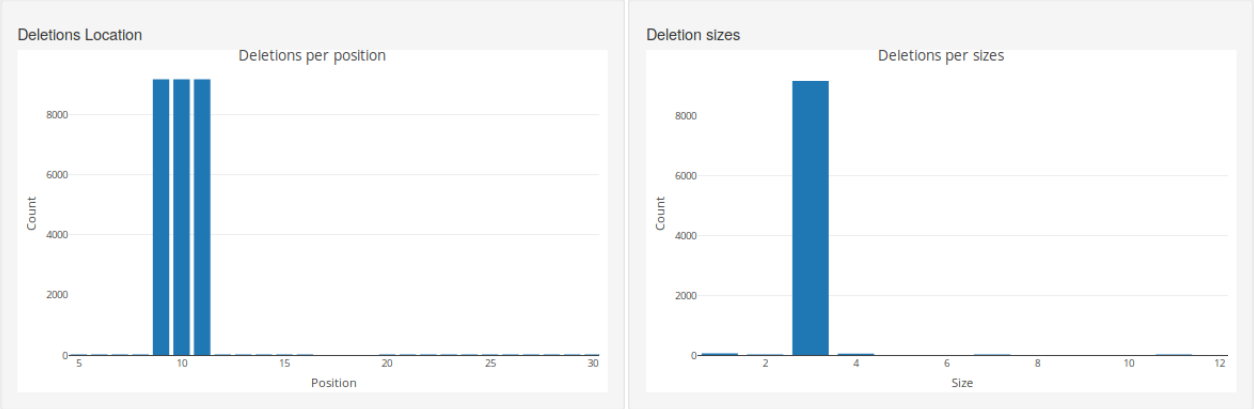


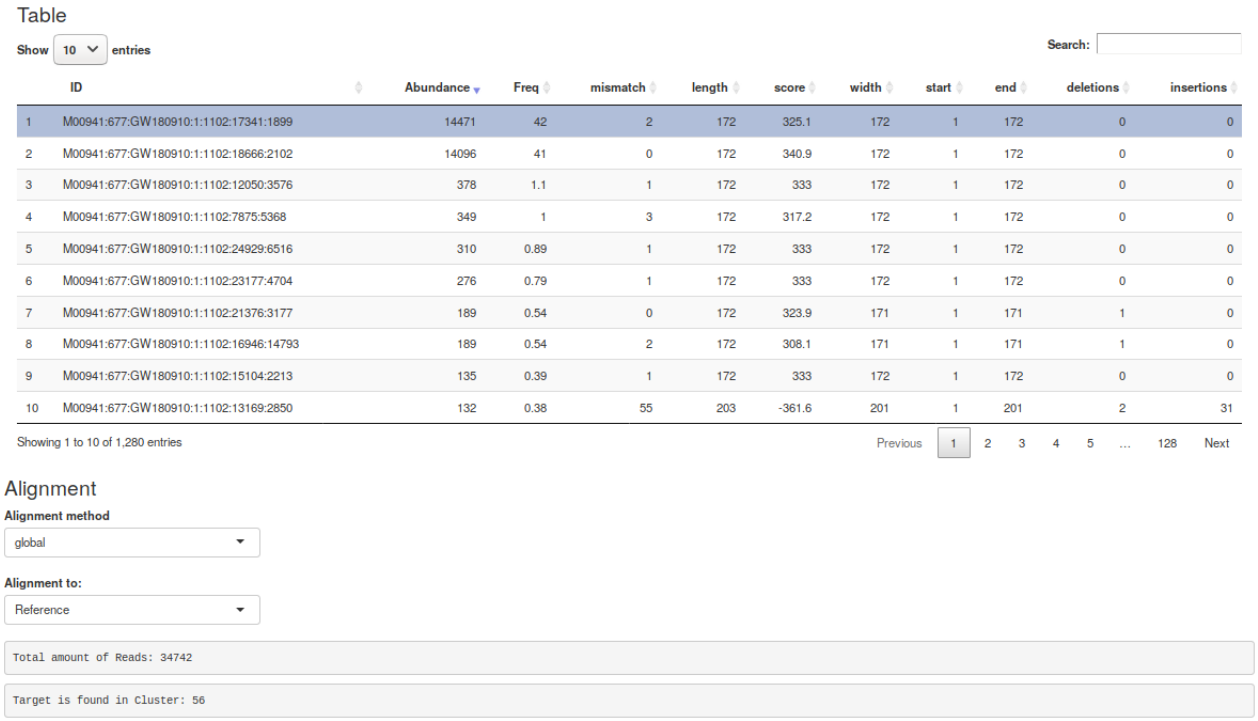
Figura 4.11: **Izquierda: Deleciones por posición. Derecha: Deleciones por tamaño.**

También se puede observar en el gráfico de *deleciones por tamaño* un claro dominio por parte de las deleciones de 3 bases frente a las deleciones de otros tamaños.

4.2.2. Experimento en muestra humana

El experimento en humanos tiene como objetivo la corrección de una mutación patogénica de una muestra tisular de un paciente. Inicialmente el paciente mostraba un mosaicismo en el que se podían observar un alelo con la mutación patogénica, un alelo sin mutación, y otro dos alelos con mutación y sin ella pero con un polimorfismo. Como añadido, al intentar corregir la mutación mediante un protocolo de CRISPR-CAS9 , deberiamos de poder observar otros alelos que contengan la corrección, por sí sola, además de otro que la contenga junto al polimorfismo. De manera que estos suponen datos muy interesantes para observar mediante la aplicación desarrollada.

En la Figura 4.12 se observan los resultados. Al introducir nuestros datos junto a un *target* equivalente a nuestra mutación corregida, ya vemos que la aplicación nos informa de que el alelo más parecido se encuentra en la entrada 56. Esto puede ser debido a un error de la aplicación, o bien que la inyección de las moleculas de CRISPR-CAS9 ha tenido muy poca penetración. Proseguimos entonces a buscar entre las distintas entradas mayoritarias (>1 % de representación) para comprobar si la mutación correctiva se encuentra entre ellas, sin embargo no conseguimos observarla. A continuación buscamos el resto de las secuencias que sí deben de estar presentes para descartar que el error provenga de la aplicación. Al hacer click en las primeras entradas ya encontramos nuestra secuencia de referencia sin mutación patogénica, además de su análoga con el polimorfismo. También observamos entre los alelos mayoritarios la mutación patogénica. Esto nos indica que probablemente la baja presencia de la mutación correctiva no provenga de un error en el proceso de análisis.



Table

Show 10 entries Search:

	ID	Abundance	Freq	mismatch	length	score	width	start	end	deletions	insertions
1	M00941:677:GW180910:1:1102:17341:1899	14471	42	2	172	325.1	172	1	172	0	0
2	M00941:677:GW180910:1:1102:18666:2102	14096	41	0	172	340.9	172	1	172	0	0
3	M00941:677:GW180910:1:1102:12050:3576	378	1.1	1	172	333	172	1	172	0	0
4	M00941:677:GW180910:1:1102:7875:5368	349	1	3	172	317.2	172	1	172	0	0
5	M00941:677:GW180910:1:1102:24929:6516	310	0.89	1	172	333	172	1	172	0	0
6	M00941:677:GW180910:1:1102:23177:4704	276	0.79	1	172	333	172	1	172	0	0
7	M00941:677:GW180910:1:1102:21376:3177	189	0.54	0	172	323.9	171	1	171	1	0
8	M00941:677:GW180910:1:1102:16946:14793	189	0.54	2	172	308.1	171	1	171	1	0
9	M00941:677:GW180910:1:1102:15104:2213	135	0.39	1	172	333	172	1	172	0	0
10	M00941:677:GW180910:1:1102:13169:2850	132	0.38	55	203	-361.6	201	1	201	2	31

Showing 1 to 10 of 1,280 entries Previous 1 2 3 4 5 … 128 Next

Alignment

Alignment method: global

Alignment to: Reference

Total amount of Reads: 34742

Target is found in Cluster: 56

Figura 4.12: **Tabla informativa de datos de muestra humana.**

Se procede con la observación de los gráficos y observamos una distribución llamativa de las posiciones de las deleciones. Como se puede observar en la Figura 4.13, existe una gran mayoría de deleciones que se encuentran entre las posiciones 100 y 150, rango de posicionamiento que coincide con el locus de anclaje de la molecula CRISPR-CAS9 . Vemos también que en el gráfico de *Deleciones por tamaño* hay una gran presencia de deleciones de gran tamaño. Procedemos a buscar entre los alineamientos de las entradas estas supuestas deleciones, y vemos múltiples entradas distintas que contienen grandes *gaps* siempre en el area donde CRISPR-CAS9 debería de haber llevado a cabo la mutación. En el anexo C se puede observar en la Figura C.3 unos alineamientos que muestran los *gaps* mencionados. Esto puede querer decir que no es que la penetración de la molécula haya sido baja, si no que el proceso de reparación por HDR no ha funcionado correctamente en este experimento.

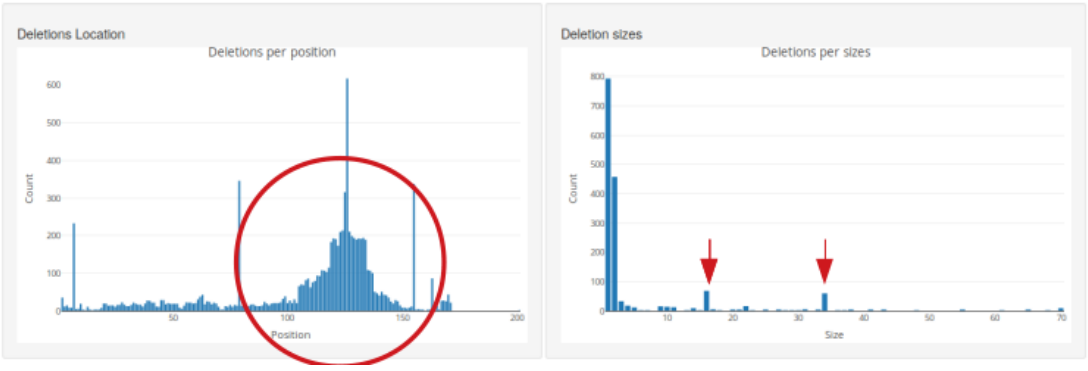


Figura 4.13: **Izquierda: Deleciones por posición, destaca la distribución de deleciones entre las posiciones 100 y 150.**

5

Conclusiones y trabajo futuro

5.1. Conclusiones

En el experimento de los ratones el hecho de que los gráficos de deleciones sean coherentes con los objetivos del experimento es un indicativo del buen funcionamiento de la aplicación. Al desaparecer la inserción que ocurría en el ratón fundador nos indica que efectivamente podía ser un error experimental proveniente de los primers, ya que de haber sido una verdadera inserción en la secuencia real mutada, al ser mayoritaria, habría sido heredado por el ratón mutante, sin embargo esa inserción desaparece por completo.

Tras estos resultados se estima que la aplicación presenta un grado de funcionamiento aceptable, pues ha permitido no sólo la correcta visualización del mosaicismo alélico del experimento con CRISPR-CAS9 en ratón, si no también identificar un posible error de diseño experimental en el caso del experimento en la muestra tisular humana. Además la confirmación de la presencia de un polimorfismo en la muestra no habría podido ser realizada a través de las herramientas actuales de visualización web, pues estas serían erróneamente calificadas como mutaciones NHEJ debidas a CRISPR-CAS9 tanto por CRISPREsso como por CRISP-ga.

Por estos motivos se estima que la aplicación supone un avance a las herramientas web disponibles actualmente.

5.2. Trabajo futuro

A pesar de que el funcionamiento de la aplicación es aceptable, existe mucho trabajo que llevar a cabo antes de hacerla pública.

Existen múltiples errores que deben de ser pulidos, como es crasheo del módulo de descarga de informes tras múltiples descargas seguidas. Es necesario también un informe de errores más claro entendibles por un usuario genérico, además de evitar el crasheo de la aplicación en caso de que estos ocurran, bien con un mayor uso de operaciones tipo *try-catch* o mediante la posibilidad de reiniciar la aplicación automáticamente en presencia de un error fatal e impredecible.

Sería enormemente conveniente la posibilidad de introducir una secuencia en alguna entrada y que la aplicación sea capaz de buscar esta secuencia entre las entradas ya generadas con ante-

rioridad. De manera que no haya que reiniciar toda la aplicación para poder variar la secuencia *Target* cada vez que se quiera buscar entre todos los clusters de forma automática.

Sería conveniente también el uso de herramientas de análisis de secuencias con mayor variedad de funciones como podría ser *Trimmomatic* [39], que es capaz de llevar a cabo todas las funciones de *cutadapt* y de *prinseq*.

Por otro lado, durante este trabajo no se ha tenido en cuenta, más allá de saber que existen, las particularidades del método de secuenciación del secuenciador MISEQ de Illumina, el secuenciador con el que se ha estado trabajando. No se conoce con exactitud cómo funciona la detección de errores propios de Illumina y por lo tanto los softwares actuales no son capaces de adaptarse a las particularidades de sus máquinas. Por este motivo comvendría llevar a cabo una correción al análisis de las secuencias teniendo en cuenta estudios llevados a cabo especificamente en las máquinas a utilizar [40].

Por último, actualmente la herramienta funciona únicamente de forma local. Alojar la herramienta en un servidor capaz de procesar y almacenar la información paralela de múltiples usuarios, de forma segura, conectados simultaneamente a través de distintos dominios web, requerirá en sí mismo una gran cantidad de trabajo.

Glosario de acrónimos

- **DNA**: Ácido Desoxirribonucleico
- **RNA**: Ácido Ribonucleico
- **CRISPR**: Repeticiones Palindrómicas Cortas Agrupadas y Regularmente Interespaciadas
- **CAS9**: CRISPR proteína asociada 9
- **HDR**: Reparación Dirigida por Homólogía
- **NHEJ**: Unión de Extremos Vecinos no-Homóloga
- **NCBI**: Centro Nacional de Información Biotecnológica.
- **BD**: Base de Datos.
- **URL**: Localizador de recursos uniforme.
- **ID**: Identidad
- **COV**: Cobertura

Bibliografía

[1] Feng Zhang, Yan Wen, and Xiong Guo. Crispr/cas9 for genome editing: progress, implications and challenges. *Human Molecular Genetics*, 23(R1):R40–R46, 2014.

[2] Luca Pinello, Matthew C. Canver, Megan D. Hoban, Stuart H. Orkin, Donald B. Kohn, Daniel E. Bauer, and Guo-Cheng Yuan. Analyzing crispr genome-editing experiments with crispresso. *Nature Biotechnology*, 34:695 EP –, Jul 2016.

[3] Marc Güell, Luhan Yang, and George M. Church. Genome editing assessment using crispr genome analyzer (crispr-ga). 30(20):2968–2970, Oct 2014. 24990609[pmid].

[4] Rehm HL. Evolving health care through personal genomics. 2017.

[5] Rehm HL. Korf B. New approaches to molecular diagnosis. pages 1511–21, Apr 2013.

[6] Zampiga V Danesi R Arcangeli V Ravegnani M Canginil Pirini F Petracci E Rocca A Falcini F Amadori D Calistri D Tedaldi G, Tebaldi M. Multiple-genepanel analysis in a case series of 255 women with hereditary breast and ovariancancer. 4:25)–267, Apr 2017.

[7] Lesende I Limongelli I Ranzani G Novara F Bonaglia M Rinaldi B Franchi F Manolakos E Lonardo F Scarano F Scarano G Costantino L Tedeschi S Giglio S Zufardi O Vetro A, Godin D. Diagnostic application of a capture based ngstest for the concurrent detection of variants in sequence and copy number as well as loh. May 2017.

[8] Ng C Ban KH Tan TW Huan PT Lee PL Chiu L Seah E Ng CH Koay ES Chng WJ Yan B, Hu Y. Coverage analysis in a targeted amplicon-based next-generation sequencing panel for myeloid neoplasms. 69:801–804, Sep 2016.

[9] Charpentier E Doudna JA. The new frontier of genome engineering with crispr-cas9. pages 801–804, Nov 2014.

[10] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018.

[11] Inc RStudio. *Easy web applications in R.*, 2013. URL: `http://www.rstudio.com/shiny/`.

[12] Peter Kelsey Ivan Adzhubei Christina A. Austin-Tse Jeffrey D. Cooney3 Heidi Anderson Ignaty Leshchiner, Kristen Alexa. Mutation mapping and identification by whole-genome sequencing. pages 1541–1548, 2012.

[13] Illumina Inc. *An introduction to Next-Generation Sequencing Technology*, 2017.

[14] A. McKenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, and M. A. DePristo. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.*, 20(9):1297–1303, Sep 2010.

[15] Tee Louis Y Wang Xiao-Gang Huang Qun-Shan Yang Shi-Hua Zhang, Xiao-Hui. Off-target effects in crispr/cas9-mediated genome engineering. 4, 2015.

[16] Alejandro A. Schäffer Jinghui Zhang-Zheng Zhang Webb Miller Stephen F. Altschul, Thomas L. Madden and David J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res.*, 25:3389–3402, 1997.

[17] Sanger F; Coulson AR. A rapid method for determining sequences in dna by primed synthesis with dna polymerase. May.

[18] Sanger F; Nicklen S; Coulson AR. Dna sequencing with chain-terminating inhibitors. *Proc. Natl. Acad. Sci. U.S.A.*, 74:5463Ű7, December 1977.

[19] Barrangou R. The roles of crispr-cas systems in adaptive immunity and beyond. 32:36Ű41, 2015.

[20] B; Aguilera Pardo, B; Gomez-Gonzales. Dna repair in mammalian cells: Dna double-strand break repair: how to fix a broken relationship. *Cellular and Molecular Life Sciences*, 66:1039Ű1056, 1970.

[21] Hadley Wickham. *tidyverse: Easily Install and Load the Tidyverse*, 2017. R package version 1.2.1.

[22] Carson Sievert. *plotly for R*, 2018.

[23] JJ Allaire, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, and Winston Chang. *rmarkdown: Dynamic Documents for R*, 2018. R package version 1.10.

[24] Marcel Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, 17(1):10–12, May 2011.

[25] H. Pagès, P. Aboyoun, R. Gentleman, and S. DebRoy. *Biostrings: Efficient manipulation of biological strings*, 2018. R package version 2.48.0.

[26] Robert Schmieder and Robert Edwards. Quality control and preprocessing of metagenomic datasets. *Bioinformatics (Oxford, England)*, 27(6):863–864, March 2011. PMID: 21278185.

[27] Erik Aronesty. Command-line tools for processing biological sequencing data. 2011.

[28] Heng Li. Toolkit for processing sequences in fasta/q formats.

[29] Illumina. Patterned flow cell technology, 2015.

[30] Hagopian R. Samorodnitsky E., Jewell B. Evaluation of hybridization capture versus amplicon-based methods for whole-exome sequencing. human mutation. 36:903–914, 2015.

[31] Ewing B; Hillier L; Wendl MC; Green P. Base-calling of automated sequencer traces using phred. i. accuracy assessment. *Genome Research*, pages 175–185, 1998.

[32] Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

[33] Hadley Wickham and Lionel Henry. *tidyr: Easily Tidy Data with spread() and gather() Functions*, 2018. R package version 0.8.2.

[34] Hadley Wickham, Romain François, Lionel Henry, and Kirill Müller. *dplyr: A Grammar of Data Manipulation*, 2018. R package version 0.7.6.

[35] Patrick Aboyoun. *Pairwise Sequence Alignments (Biostrings)*, 2019.

[36] Christian D Needleman, Saul B. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–53, 1970.

[37] Michael S. Smith, Temple F. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195Ű197, 1981.

[38] Brudno M; Malde S; Poliakov A; Do CB; Couronne O; Dubchak I; Batzoglou S. Glocal alignment: finding rearrangements during alignment. *Bioinformatics*, 19:54Ű62, 2003.

[39] Anthony M. Bolger, Marc Lohse, and Bjoern Usadel. Trimmomatic: a flexible trimmer for illumina sequence data. *Bioinformatics*, 30(15):2114–2120, Aug 2014. 24695404[pmid].

[40] Christopher Quince, Umer Z. Ijaz, William T. Sloan, Melanie Schirmer, Neil Hall, and Rosalinda D'Amore. Insight into biases and sequencing errors for amplicon sequencing with the Illumina MiSeq platform. *Nucleic Acids Research*, 43(6):e37–e37, 01 2015.

A

Manual de utilización

A.1. Mosaic Finder. (MoFi)

A.1.1. Getting Started.

The main purpose of this app is to be accesible via web domain, but for the time being the app must be run locally in order to function properly.

This instructions will attempt to aid in the deployment of the app in your local system.

Prerequisites

- Linux operating system
- Compilers:
  - R
  - perl
  - python

Installing

Clone this repository into your system:

```
$ git clone https://github.com/irycisBioinfo/CrisPRAL.git
```

or download .zip and unpack.

Testing

Load app.R file through a R compiler.

- You can use either RStudio by opening the file and running the app through the 'Run App' option at the top right corner of the script editor.
- Sourcing directly the file through RStudio's console.

```
source("path/to/file/app.R")
```

- Executing the file through your terminal window

```
$ Rscript path/to/file/app.R
```

When executing through the terminal window you will have to manually copy the ip adress displayed and paste it in your browser of preference.

You will find yourself with the main page presenting the mosaic finder functionality. Enter the obligatory fields, Read 1, Read 2, and the reference. You can find demo files for these fields in */Trials* directory.

The rest are optional options but demo files are presented aswell for the Adapters filtering and the PCR Primers trimming.

Known Issues

Some issues are known to occur and do not display an appropriate error message to the user.

- If for any reason, all reads of the Read 1 and Read 2 files are filtered out, the program will halt and the browser will require to be re-loaded.
- Download of automated report may produce an error after several attempts at downloading, the browser will again be require to be re-loaded.

B

Manual del programador

En el presente manual se presentan ciertos códigos junto con la Figura B.1 que esquematiza el flujo entre cada script neceserio para la apropiada ejecución de la aplicación web.

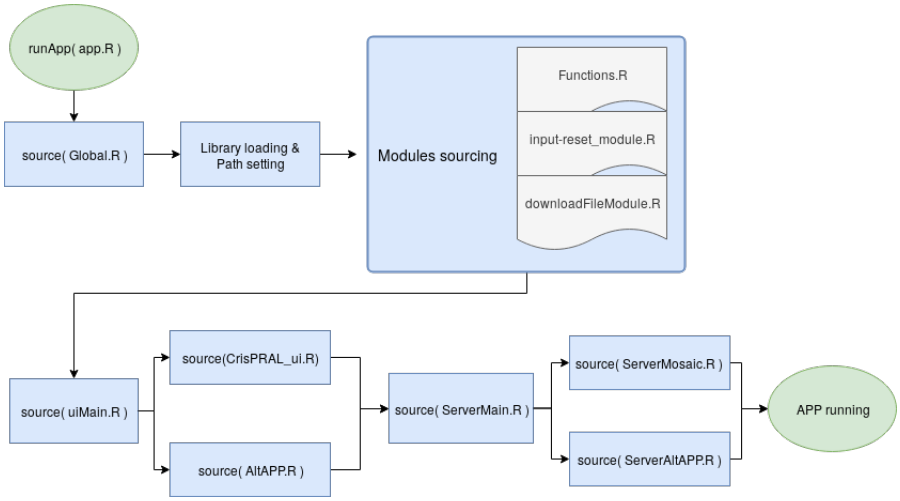


Figura B.1: **Diagrama de flujo que esquematiza las relaciones entre los archivos ejetubles de la aplicación.**

Cada uno de estos scripts se pueden encontrar en un repositorio en GitHub.

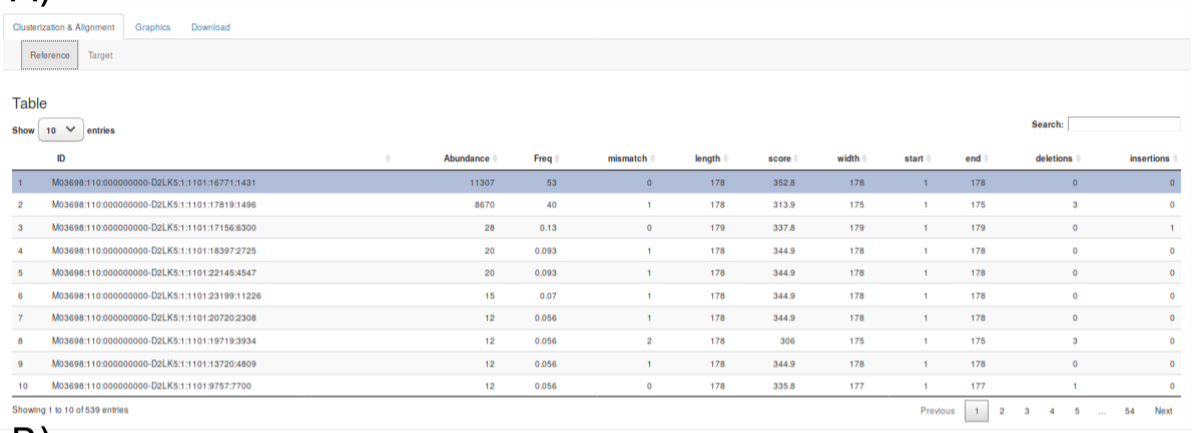
- **app.R:** Script que carga los demás en el orden designado.
- **Global.R:** Carga o instala paquetes de R necesarios, determinina las variables de PATH y carga otros módulos asistenciales.
- **Functions.R:** Contiene funciones no específicas de Shiny requeridas para ciertas operaciones de la sección de server.
- **input-reset_module.R:** Módulo de Shiny que permite el reinicio de entradas de ficheros en la sección de Interfaz (UI)
- **downloadFileModule.R:** Módulo para la descarga de archivos.

- **uiMain.R:** Script que carga las interfaces de las aplicaciones.
- **CrisPRAL_ui.R:** Carga la interfaz de la aplicación principal de Mosaic Finder.
- **AltAPP.R:** Carga la interfaz de una aplicación alternativa, este caso se ejemplifica con Lazy Panel Filter.
- **ServerMain.R:** Script que carga los servidores de Shiny donde se operan con los datos.
- **ServerMosaic.R:** Script que carga el servidor de Mosaic Finder.
- **ServerAltAPP.R:** Script que carga el servidor de una aplicación alternativa, en este caso se ejemplifica con Lazy Panel Filter.

C

Anexo Figuras

A)



B)

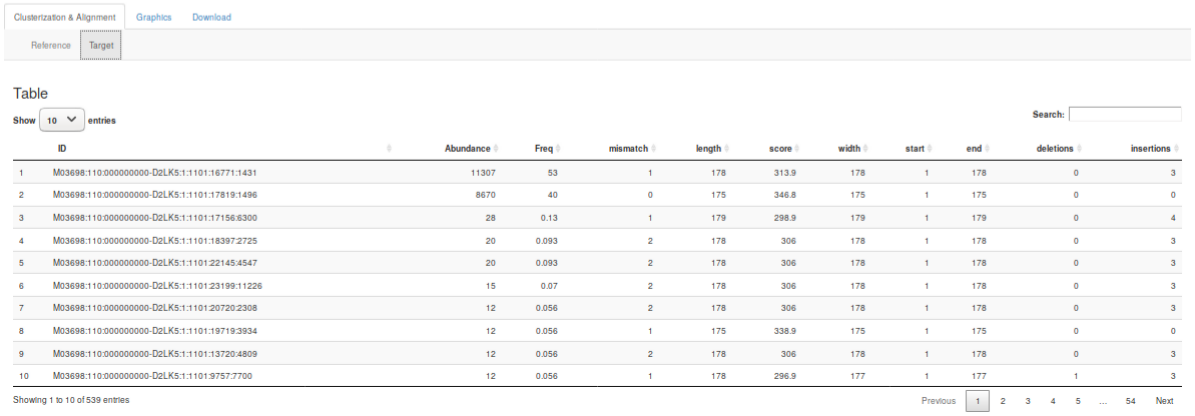


Figura C.1: **Pantallazo de Interfaz final comparando: A) tabla informativa generada a partir de la referencia frente a B) la generada utilizando la secuencia Target.**

```
Output created: reports.pdf
Warning: Error in : no se puede ubicar un vector de tamaño  11.0 Gb
  [No stack trace available]
```

Figura C.2: **Error tras multiples descargas de informe automatico consecutivas.**

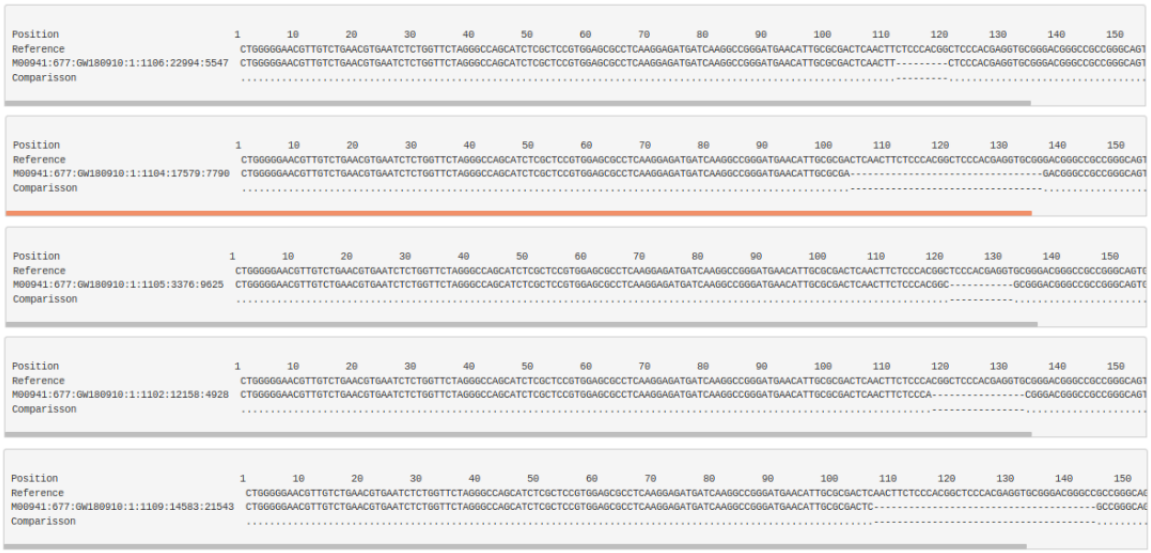


Figura C.3: **Alineamientos de entradas con gaps en el area de anclaje de CRISPR-CAS9 . Datos de muestra humana.**

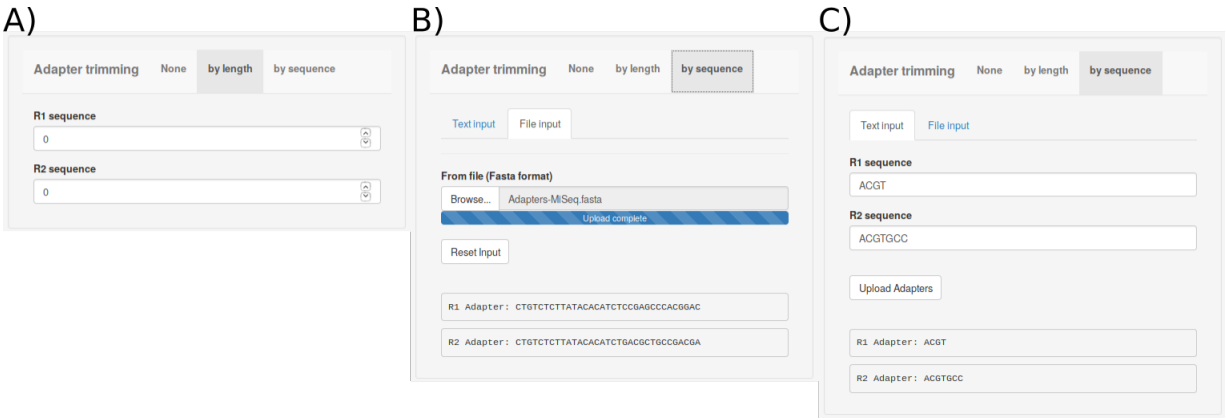


Figura C.4: **Opciones de Trimming de Adaptadores desplegadas.**

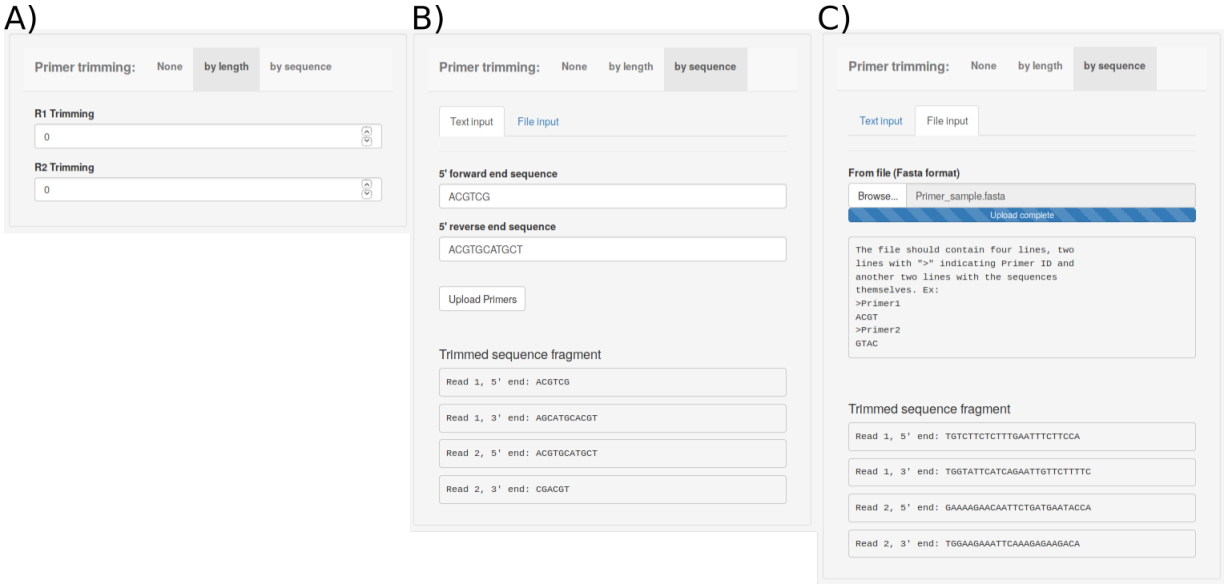


Figura C.5: **Opciones de Trimming de Primers desplegadas.**